

VOVVOV

Nummer 1

19. august 1987

I dette nummer:

Hvad er »Vovvov«?
Hvorfor indeholder ISO 8859/1 Intet Æ?
Status på ISO 8859.
Brug af *netnews* på uucp-nettet.
ORACLE, version 5
Parametre til login
UUCP-adresser
DIAB C-oversætter
How To Write a UNIX Daemon.
Nye »hak«-sekvenser til System V

1. Hvad er »Vovvov«?

Det blad du i øjeblikket læser er et forsøg på at øge kommunikationen af tekniske informationer imellem afdelingerne i DDE.

Det er tanken at Vovvov skal være et internt teknisk tidsskrift i DDE. I Vovvov vil folk kunne skrive tekniske artikler til hinanden for på den måde at gøre andre delagtige i erfaringer man har gjort med anvendelse af UNIX og andre Supermax produkter.

Typiske artikler i Vovvov kan være:

- beskrivelser af nyttige shell-scripts
- typiske programmeringsfejl
- beskrivelser af nyttige udokumenterede egenskaber ved diverse programmer
- standardiseringstiltag
- nye produkter til afprøvning internt i DDE

Det skal understreges at Vovvov skal være et *teknisk* tidsskrift. Oplysninger om firmapolitik, afdelingsstruktur, kritik af andre afdelingers politik m.m. vil kun undtagelsesvis blive optaget. Så selv om alle har lov til at skrive i Vovvov, vil der alligevel blive udøvet en vis form for censur. Anonyme indlæg optages ikke, og alle meningstilkendegivelser står for forfatterens egen regning.

Et åbent spørgsmål er graden af fortrolighed i Vovvov. Det er min personlige mening at man i Vovvov skal have lov til at orientere om endnu ikke frigivne produkter; men det betyder på den anden side, at man skal være meget påpasselig med ikke at lade udenforstående få fat i ens kopi af bladet. Jeg vil gerne høre andres meninger om dette.

Vovvov vil udkomme efter behov, dvs. når der er artikler nok til at forsvare en udsendelse.

Det bliver undertegnede der kommer til at fungere som redaktør på Vovvov. Jeg vil gerne indtrængende opfordre alle til at sende mig indlæg til bladet. For at minimere mit arbejde, bedes I venligst sende mig artiklerne på magnetisk form, fx via uucp-nettet, hvor min postadresse er ct. Fotokopier af artikler kan dog også optages, som det fx fremgår af nærværende nummer.

Hvabehar? — Navnet? — Jo, altså, øh ... DDE har lissom en vis forkærlighed for hunde, ikk'?

Med venlig hilsen
Claus Tøndering
Basisprogrammelaafdelingen

2. Hvorfor indeholder ISO 8859/1 intet Œ?

Hvis en franskkyndig studerer det internationale tegnsæt, ISO 8859/1, som anvendes på Supermax, vil han eller hun undre sig over fraværet af den franske ligatur¹ Œ og œ. Hvorfor findes dette tegn ikke i tegnsættet?

Det har været meget svært for mig at efterforske dette, men for nylig fik jeg kontakt med Tim Lasko fra DEC som har været involveret i standardiseringsarbejdet, og han gav følgende forklaring:

De allerførste forslag til 8859/1 indeholdt faktisk Œ og œ i position D7 og F7. Mange følte at disse to tegn var vigtige for fransk. Men i 1985 erklærede AFNOR (den franske standardiseringsorganisation) at disse tegn kunne fjernes, eftersom de strengt taget ikke var en del af det franske alfabet, men blot en meget hyppigt benyttet trykningsform. Da standardens formål var at være effektiv i databehandling, og da Œ altid blev behandlet som de adskilte bogstaver O og E, var der ingen grund til at have en separat kode for ligaturen.

Dette bevirkede at det næste forslag til 8859/1 havde D7 og F7 uspecificerede, og i den endelige forslag til 8859/1 blev disse positioner fyldt med de nuværende multiplikations- og divisionstegn. Indføjelsen af disse to tegn var nødvendig for at undgå at forskellige firmaer brugte disse to positioner til private tegn, hvilket ville føre til små men vanskelige inkompatibiliteter. Multiplikations- og divisionstegnet blev valgt som et kompromis mellem flere kandidater.

I slutningen af 1986, da 8859/1 var til endelig afstemning i ISO, stemte Frankrig imod. Nu ville de nemlig alligevel have Œ og œ med! Det viste sig at det franske P&T først på nuværende tidspunkt var blevet medlem af AFNOR's kodningskomité, og dette nye medlemsskab havde bevirket at stemningen i AFNOR nu var ændret, idet det franske P&T gerne ville have ligaturerne med.

1. En ligatur er en typografisk sammenskrivning af to tegn.

Men nu var det for sent. Det endelige forslag til 8859/1 var fastlagt, og det blev vedtaget med kun to nej-stemmer (Frankrig og Tjekkoslaviet).

Ak, ja!

ct [BP]

3. Status på ISO 8859

Udover vores ISO 8859/1 findes der en række andre 8859-standarder:

8859/2 er et udvidet ASCII-alfabet til brug i Østeuropa. Det dækker sprog som tysk, tjekkisk, slovakisk, posk, serbokroatisk (latinsk variant), ungarsk, albansk, rumænsk og slovensk. Denne standard er vedtaget.

8859/3 er et udvidet ASCII-alfabet til brug i bl.a. middelhavsområdet. Det dækker sprog som afrikaans, catalansk, fransk, italiensk, maltesisk, spansk, tyrkisk(OBS!), esperanto m.m. Denne standard er til første afstemning.

8859/4 er et udvidet ASCII-alfabet til brug i nordeuropa. Det dækker sprog som finsk, dansk(!), norsk, grønlandsk, lappisk, estisk, lettisk, litauk og tysk. Men til trods for at standarden hævdes at dække dansk og norsk indeholder den intet Å! Denne standard er til første afstemning.

8859/5 er ASCII-alfabetet udvidet med det kyrilliske alfabet. Det dækker sprog som russisk, ukrainsk, hviderussisk, bulgarsk, serbokroatisk (kyrillisk variant) og makedonsk. Denne standard afventer input fra Sovjetunionen.

8859/6 er ASCII-alfabetet udvidet med det arabiske alfabet. Denne standard er vedtaget.

8859/7 er ASCII-alfabetet udvidet med det græske alfabet. Denne standard er til første afstemning.

8859/8 er ASCII-alfabetet udvidet med det hebraiske alfabet. Denne standard er endnu kun et forslag.

Ovennævnte standarder og standardforslag (bortset fra nummer 7 og 8) kan beses hos undertegnede.

ct [BP]

4. Brug af netnews på uucp-nettet

Personer der hænger på uucp-nettet kan hente diskussionsinlæg, programmer, nyheder m.m. fra det internationale uucp-net. Vi er i gang med en forbedring af denne service, men p.t. fungerer den som følger:

Første trin er at få fat på sidste nye liste over emnegrupper fra DIKU. Dette sker ved at sende et brev til archive@diku. Dette brev skal kun indeholde én linie, nemlig følgende:

Subject: groups

Dette bevirker at man efter nogen tid modtager en emneliste med elektronisk post fra DIKU. DIKU finder selv ud af afsenderadressen.

Hvis man nu fra denne liste fx ønsker at vide mere om emnet »comp.lang.ada« sender man et nyt brev til archive@diku. Dette brev skal også kun indeholde én linie, nemlig følgende:

Subject: lines comp.lang.ada

Dette bevirker at man efter nogen tid modtager en liste over artikler inden for dette emne. Alle artikler har et nummer og en emne-linje, som er angivet i listen. Hvis denne emne-linje starter med »Re:« er der tale om en kommentar til et tidligere (muligvis ikke længere eksisterende) indlæg. Hvis man fra artikellisten ønsker at se artiklerne 304, 306 og 310, sender man et sidste brev til archive@diku. Dette brev skal indeholde to linier:

Subject: articles comp.lang.ada
Names: 304 306 310

Efter nogen tid vil man så modtage artiklerne med elektronisk post.

ADVARSEL: Hvis alle i DDE nu begynder at hente filer fra DIKU bliver vores uucp-net hurtigt overbelastet. Så vær lidt hensynsfulde! Eller endnu bedre: Vent med at bruge faciliteten til vi har fået forbedret vort system, så vi får artiklerne over på vores egen maskine.

ct [BP]

5. ORACLE, version 5

Portningen af den nye version af ORACLE er i fuld gang i BP, og vi vil med dette indlæg give en orientering om denne version.

Vi har for noget tid siden portet version 5.0, og denne version findes til test i nogle få eksemplarer hos bl.a. AS, LISDA og på Mother. Vi gør imidlertid opmærksom på, at denne version *ikke* vil blive frigjort som et produkt, og at den kun vil kunne leveres efter aftale med BÖ. Vores officielle udgave af ORACLE 5, vil blive den vi porter i øjeblikket som benævnes 5.1, og denne vil være leveringsklar i fjerde kvartal i år. Manualer til ORACLE 5.1 kan fås hos AM.

Der er følgende at glæde sig til ved ORACLE 5.1:

- Sorteringer er ændret til at foregå i hovedlageret fremfor i temporære tabeller, hvilket giver en betydelig hastighedsforøgelse for operationer som ORDER BY, GROUP BY, SELECT DISTINCT, CREATE INDEX m.fl.
- Der er iøvrigt foretaget andre optimeringer og der er en generel forbedring af performance.
- Der er indført en facilitet til fetch eller insert af flere rækker på en gang, hvilket bl.a. har gjort export/import hurtigere.
- Context area (cursor størrelsen) udvides automatisk.

ORACLE har fra version 5 adskilt kernen (med ior, exp, imp, ods, aij) fra utilities, således at disse ikke længere har samme nummer som kernen. I vores første release af ORACLE 5.1 vil følgende være til rådighed:

SQL*Plus

Dette produkt er afløseren for ufi, og vi har version 2.0 af denne.

SQL*Forms

SQL*Forms er en kraftigt udvidet iaf, idet der er kommet faciliteter som screen painter, form-, block- og rowlevel trigger, redifinerbare funktionstaster, makroer m.v. Vi har version 2.0.

SQL*Report

Dette lyder flot; men er (foreløbig) blot et nyt navn for de kendte rpt og rpf.

PRO*C

Nyt navn for pcc til C.

PRO*SQL

Nyt navn for hli. PRO*SQL er udvidet med den tidligere nævnte mulighed for at lave fetch/insert af flere rækker på en gang.

Følgende produkter vil være til rådighed senere:

SQL*NET

Dette giver mulighed for distribuerede forespørgsler, dvs. select på tværs af maskiner. I første omgang vil det formentlig være på Supermax'er sammenkoblet med Ethernet med TCP/IP protokollen.

SQL*CALC

Dette er ORACLE's spreadsheet produkt, der så vidt vi ved minder noget om lotus 1-2-3 udvidet med databaseadgang.

SQL*GRAPH

SQL*GRAPH er en udgave af SQL*PLUS, hvor data kan præsenteres på grafisk form.

SQL*MENU

ORACLE's menusystem.

EASY*SQL

Dette er en skærmorienteret SQL fortolker, hvor valg af søgekriterier m.v. sker med udpegninger i pop-up menuer.

Dansk

De vigtigste slutbrugerprodukter vil forekomme i danske udgaver. I første omgang vil det kun dreje sig om afviklingsdelen af SQL*FORMS.

Det skal bemærkes, at der endnu ikke er sat terminer på færdiggørelsen af ovennævnte.

be [BP]

6. Parametre til login

En nyttig med lidet kendt egenskab ved login er muligheden for at overføre parametre til brug for fx .profile-filen.

Kort fortalt er princippet følgende: Når man skal logge på, plejer man at taste sit *login id*. Efter dette kan man imidlertid taste yderligere et antal ord. Disse ord vil blive gemt i environment-variablene L0, L1, L2, osv.

Normalt logger jeg på maskinen således:

```
login: ct
```

Men hvis jeg i stedet skriver:

```
login: ct ampex /usr/pip
```

vil environment-variablen L0 få tildelt værdien »ampex«, og L1 vil få værdien »/usr/pip«, hvilket jeg så fx kan benytte i min .profile-fil til at styre forskellige ting.

Typiske anvendelser kan være kørsel af *terminology* med passende parametre, eller ændring af home-directory til brug for forskellige projekter jeg arbejder på.

Læs mere i »System V Reference Manual, section 1b« under *login*(1).

ct [BP]

7. UUCP-adresser

På grund af nogle ændringer på det globale uucp-net bedes I venligst bemærke følgende: Vor adresse udadtil må kun indeholde små bogstaver. Hvis I derfor tidligere har opgivet jeres adresse som »xx@dde.UUCP«, skal I fremover benytte »xx@dde.uucp«.

ct [BP]

8. DIAB C-oversætter

Vi har fået en evalueringversion af en C-oversætter til 68020, fra det svenske DIAB.

Hvoved adskiller den sig fra vores nuværende C-oversætter, og hvad udmærker den sig ved?

Nogle af de annoncerede forbedringer er:

- hurtigere oversættelse
- hurtigere (og om muligt også kortere) kode
- bedre fejlrapportering

Hvem fristes ikke af disse ting, når der samtidig loves, at den oversætter samme C-dialekt som vores gamle oversætter?

Denne nye oversætter (dcc) er på prøve rundt om i huset, for at finde ud af om den lever op til det lovede, og om den kan klare nogle af de problemer vi har haft med den nuværende oversætter.

De foreløbige resultater ser lovende ud,

- ca 20% hurtigere kode (end vores nuværende 68020 oversætter),
- fejlmeldinger der er til at forstå,
- en del af vores kendte fejl i gamle oversætter, findes ikke her.

Vi arbejder videre og håber at den kan bruges.

hf [BP]

9. How To Write a UNIX Daemon

De følgende sider indeholder en fotokopi af en artikel fra USENIX's blad »;login:« (herligt navn, ikke?). USENIX er en amerikansk UNIX-brugergruppe. Artiklen vil sikkert være af interesse for mange, hvorfor den gengives her.

ct [BP]

10. Nye »hak«-sekvenser til System V

Sidste side i dette blad indeholder en beskrivelse af nye styresekvenser til System V.

I beskrivelsen bruges tegnene # og \$ til at betegne tal som er parametre til styresekvenserne.

pe [BP]

;login:

How To Write a UNIX Daemon

Dave Lennert

Hewlett-Packard Company
hplabs!hpda!davel

ABSTRACT

On UNIX systems users can easily write daemon programs that perform repetitive tasks in an unnoticed way. However, because daemon programs typically run outside a login session context and because most programmers are unfamiliar with designing a program to run outside this context, there are many subtle pitfalls that can prevent a daemon from being coded correctly. Further, the incompatibilities between various major UNIX variants compound these pitfalls. This paper discusses these pitfalls and how to avoid them.

Daemon: runs around in the shadows (background) doing devilish deeds.

found in some daemon source code

Introduction

A daemon is a program which performs periodic tasks in such a manner that it is normally unnoticed by users.

Some daemons run constantly, waiting for a significant event. Examples include `init` which respawns login sessions (`gettys`) as they end, `cron` which launches programs at specified times, and `sendmail` which listens on a socket for incoming mail messages.

Other daemons are launched periodically and terminate after completing one execution of their task. Such daemons include the `uucp` file transfer utility, `uucico`, which can be launched as a login shell when a remote machine logs in, `calendar` which is launched nightly by `cron` to examine users' calendars and mail them notification of upcoming events, and various `mail` handling utilities which allow the user's shell to continue while the collected mail message is delivered asynchronously.

Daemon programs are very easy to write in the UNIX environment. They can be written by casual users and launched periodically via the `at` command or, on System V, by a user's personal `crontab` file, or perhaps at each login via `csh`'s `.login` command file. System administrators write daemons whenever they recognize a particular administrative task is becoming routine enough to be handled automatically.

However, daemon programs appear easier to write correctly than they really are. This is because there are many quirks and side effects of UNIX which are automatically taken care of in a login session context but not in a detached, daemon program. The `init`, `getty`, `login`, and shell programs oversee such functions as setting up user ID's, establishing process groups, allocating controlling terminals, and managing job control.

If a daemon process is launched outside a login session (e.g., via `/etc/rc` or a similar function during system startup) then it needs to manage these functions itself explicitly. If a daemon process is launched from within a login session (e.g., as a background command from a login shell) then it needs to undo much of what the login process sequence has done. In order to code a daemon robustly, both concerns must be addressed.

This paper discusses these concerns and the methods for addressing them. Note that all the example coding fragments lack necessary error condition checking or handling; such handling should, of course, be added to any real daemon.

;login:

Programming Rules

The following is a set of programming rules which avoid several subtle pitfalls. A discussion of each pitfall is also given along with the rule.

Make immune to background job control write checks.

On systems which support 4.2BSD style job control, daemons which attempt I/O to their controlling terminal will stop if they were launched from `csh` in the background (with `&`). The real way around this is to disassociate yourself from your controlling terminal (see below). In some cases though, the daemon will want to perform some setup checks and output error messages before it loses its controlling terminal.

There is no way to allow a background process to read from its controlling tty. However, output can be reliably performed if the calling process ignores the SIGTTOU signal, as in:

```
#ifdef SIGTTOU
signal(SIGTTOU, SIG_IGN);
#endif
```

For safety, it is probably a good idea to ignore the other stop signals as well, as in:

```
#ifdef SIGTTIN
signal(SIGTTIN, SIG_IGN);
#endif

#ifdef SIGTSTP
signal(SIGTSTP, SIG_IGN);
#endif
```

Ignoring SIGTTIN also has the side effect of causing all background attempts to read from the controlling terminal to fail immediately and return the EIO error.

Close all open file descriptors, especially stdin, stdout, stderr.

Do not leave stray file descriptors open. More importantly, if any of the file descriptors are terminal devices then they must be closed to allow proper reset of the terminal state during logout (see below). The typical code sequence is:

```
for (fd = 0; fd < _NFIDE; fd++)
    close(fd); /* close all file descriptors */
```

Disassociate from your process group and controlling terminal.

Daemons launched during a login session inherit both the controlling terminal and the process group of that session (or, in the case of job control, of that job within the session).

As long as the daemon is still in the process group associated with a controlling terminal it is subject to terminal-generated signals (such as SIGINT or SIGHUP). As long as the daemon still has a controlling terminal it is subject to job control terminal I/O restrictions on systems which support job control.

Further, while the daemon remains in the original process group in which it started, it is subject to any signals sent to that process group by another program via `kill(2)`.

One way to prevent the daemon from receiving these "unintended" signals is simply to ignore all signals. However, this means that the signals cannot be used by the daemon for other purposes (such as rudimentary interprocess communication). Also, this approach is insufficient because there are some signals which a process cannot ignore (for example, SIGKILL or SIGSTOP).

A better approach is for the daemon to disassociate itself from both the controlling terminal and from the process group which it inherited. On 4.2BSD systems, the former can be performed via the `TIOCNOTTY ioctl(2)` and the latter via `setpgrp(2)`. Under AT&T UNIX, `setpgrp(2)` performs both functions.

;login:

However, (under AT&T UNIX) in order for `setpgrp(2)` to have its desired effect, this must be the first time the process has called `setpgrp(2)`; that is, the process must not already be a process group leader. (A process group leader is a process whose process group ID is equal to its process ID.) Since a program has no control over the process which `exec(2)`'d it, it must `fork(2)` to ensure that it is not already a process group leader before calling `setpgrp(2)`. (This is especially important if the daemon is launched from a `csch` which supports job control since `csch` automatically makes its children process group leaders. But this also happens, for example, when an imprudent user launches a daemon from a login shell via the `exec` command.)

In order to prevent locking up a user's terminal when a daemon is started (i.e., without '&'), the daemon usually `fork(2)`'s anyway and runs in the child while the parent immediately `exit(2)`'s without waiting for the child. This causes the shell to believe that the daemon has terminated.

A typical code sequence would be:

```
if (fork() != 0)
    exit(0);      /* parent */
/* child */
#ifdef BSD
setpgrp(0, getpid()); /* change process group */
if ((fd = open("/dev/tty", O_RDWR)) >= 0) {
    ioctl(fd, TIOCNOTTY, 0); /* lose controlling terminal */
    close(fd);
}
#else /* AT&T */
setpgrp(); /* lose controlling terminal & change process group */
#endif
```

Do not reacquire a controlling terminal.

Once the daemon is a process group leader without a controlling terminal (having called `setpgrp(2)` as described above) it is now potentially capable of reacquiring a controlling terminal. If it does, other processes (for example, logins) will not be able to acquire the terminal correctly as their controlling terminal.

(Interestingly, this problem does not exist under 4.2BSD. Unlike AT&T UNIX, where a terminal can only be acquired as a controlling terminal if it is not already a controlling terminal, 4.2BSD allows a process to join an already allocated controlling terminal and its process group. Basically, the process merges with the already established process group.)

The symptoms of this problem are somewhat subtle. Since `getty` and `login` are not able to acquire a controlling terminal, the special file, `/dev/tty`, cannot be successfully opened. Because of this, the `getpass(3)` routine, used by `login` to obtain the user's password, fails without ever printing the `Password:` prompt. All login attempts for accounts with passwords silently fail without ever prompting for a password. Login attempts for accounts without passwords succeed (because `getpass(3)` is never called), however the login shell does not have a controlling terminal. Terminal input and output still succeeds (via `stdin`, `stdout`, and `stderr`), but any keyboard signals are not sent to the processes spawned during this login session. Instead the signals are sent to the process which acquired this terminal as its controlling terminal (the daemon) and its descendants.

For this reason the daemon program must ensure that it does not re-acquire a controlling terminal.

On 4.2BSD systems, a new controlling terminal can only be acquired by a process with a process group ID of zero. After calling `setpgrp(2)` to set its process group ID equal to its process ID, the daemon cannot re-acquire a controlling terminal.

;login:

Under AT&T UNIX, a new controlling terminal is acquired whenever a process group leader without a controlling terminal opens a terminal which is not already the controlling terminal for another process group. On such systems the daemon can reacquire a controlling terminal when opening, say, */dev/console*, to perform logging or error reporting. Even if the daemon subsequently closes the terminal it still possesses it as a controlling terminal. There is no way to relinquish it since subsequent `setpgrp(2)` calls are ineffective. (`setpgrp(2)` has no effect if the caller is already a process group leader.) Therefore the acquisition must be prevented.

One simple way to prevent the acquisition of a new controlling terminal is to `fork(2)` yet another time *after* calling `setpgrp(2)`. The daemon actually runs in this second child and the parent (the first child) immediately `exit(2)`'s. However, on AT&T UNIX when the parent (first child) terminates, the `SIGHUP` signal is sent to the child since the parent is a process group leader. Thus, the parent must ignore `SIGHUP` before `fork(2)`'ing the second child otherwise the child will be killed. (The ignored setting is inherited by the child.) The final side effect of the terminating (process group leader) parent is to set the process group of the child to zero. The daemon (second child) now has no controlling terminal, it is in a new (zero) process group which is immune to signals from the tty driver, and it cannot acquire a new controlling terminal since it is not a process group leader.

Thus the typical code sequence becomes:

```
if (fork() != 0)
    exit(0);          /* first parent */

/* first child */
setpgrp(); /* lose controlling terminal & change process group */
signal(SIGHUP, SIG_IGN); /* immune from pgrp leader death */
if (fork() != 0)      /* become non-pgrp-leader */
    exit(0);          /* first child */

/* second child */
```

Do not "hold" open tty files.

Even after ensuring that the daemon will not re-acquire a controlling terminal when a terminal device is opened, there is a further concern:

Terminal state settings, such as BAUD rate and signal character definitions, are only reset to the default state when the last process having the terminal open finally closes it. Thus, if the daemon has a terminal open continuously, then the last close never happens and the terminal settings are not reset at logout.

Typical examples of terminal files held open by a daemon are *stdin*, *stdout*, *stderr*, and */dev/console*.

It's probably best to log errors and status messages to a disk file rather than a terminal. However, when terminal logging is desired, the "correct" method is to hold the terminal open only long enough to perform a single logging transaction. Note that this logging transaction still represents a window of time during which a logout would not reset the terminal state.

4.2BSD systems have a further problem which makes this suggestion mandatory. Whenever a new login session is initiated via `getty` or similar routine, the `vhangup(2)` system call is invoked to prevent any existing process from continuing to access the login terminal. This results in read and write permissions being removed from any currently open file descriptor which references the login terminal; this affects all processes regardless of user ID. Therefore, daemons which access a terminal that is also used for regular login sessions, must reopen it whenever access is desired. If a file descriptor for such a terminal is continuously held open, it is very likely that `vhangup(2)` will quickly destroy its usefulness.

To determine if an unknown file descriptor is a terminal device use `isatty(3)`.

;login:

Change current directory to '/

Each process has a current working directory. The kernel holds this directory file open during the life of the process. If a process has a current directory on a mounted file system, the file system is "in use" and cannot be dismounted by the administrator without finding and killing this process. (The hard part is finding the process!) Unless a process explicitly alters this via `chdir(2)`, it inherits the current directory of its parent. When launched from an interactive shell, the current directory will be whatever the user has most recently selected via the `cd` command.

Because of this, daemons should adopt a current directory which is not located on a mounted file system (assuming that the daemon's purpose allows this). The root file system, '/', is the most reliable choice. The simple call is:

```
chdir("/");
```

Reset the file mode creation mask.

A file mode creation mask, or *umask*, is associated with each process. It specifies how file permissions are to be restricted for each file created by the process. Like the current directory, it is inherited from the parent process and remains in effect until altered via `umask(2)`. When launched from an interactive shell, the *umask* will be whatever the user has most recently selected via the `umask(1)` command.

A daemon should reset its *umask* to an appropriate value. The typical call would be:

```
umask(0);
```

Other attributes to worry about.

The environment attributes discussed above are the primary ones to worry about, but the list is not exhaustive. Any attribute inherited across an `exec(2)` system call is of concern. Some other calls to be cautious of are the nice priority value (see `nice(2)`), the time left until an alarm clock signal (see `alarm(2)`), and, on 4.2BSD systems, the signal mask and set of pending signals (see `sigvec(2)`). However, these are less likely to be *accidentally* set "wrong."

Interactions With `init`

The system initialization process, `init`, is responsible for directly or indirectly starting all processes on the system (with the exception of kernel processes such as the swapper or pageout process). On many versions of UNIX, `init` keeps track of all processes which it directly spawned and it can optionally respawn them if they die or it can kill them when changing to a new system *run state* (or *level*). Under AT&T UNIX, the `/etc/inittab` file specifies the programs `init` should spawn in which run levels and whether or not they should be automatically respawned when they die. (Note that this file differs in both format and capabilities between System III and System V.)

Historically, system daemon programs are launched by the `/etc/rc` shell script which `init` launches when moving the system from the single user run state to multi-user mode.

Some system administrators now prefer to launch daemons directly from `init` by placing the appropriate commands in `/etc/inittab`. They rely on `init` respawning the daemon should it inadvertently die and on `init` killing the daemon during system state changes.

Note that the respawning and terminating capabilities of `init` depend on the spawned program not terminating prematurely. The above programming rules, however, suggest that daemons should immediately `fork(2)` and have the original process `exit(2)`. If launched from `/etc/inittab`, this procedure would cause `init` to believe that the daemon was no longer running and hence it would not terminate the daemon during state changes and would instead immediately relaunch the daemon (if automatic respawn were requested). This procedure thus defeats both the respawning and terminating capabilities provided by `/etc/inittab`.

;login:

What can be done to correct this? The only solution is to prevent the daemon from following the above procedure if it is launched from */etc/inittab*.

One tempting approach is for the daemon to retrieve the process ID of its parent immediately using `getppid(2)` and, if it is `init`'s process ID (1), skip the problematic code. However this is not perfectly reliable since any process whose original parent has terminated assumes `init` as its new parent. If a daemon is launched interactively from a user's shell, the shell might subsequently terminate before the daemon has executed the `getppid(2)` call. In short, there is a race condition. However, for practical purposes, this is a quick and easy way to solve the problem.

Another approach is to pass a command line flag to the daemon indicating whether the daemon is being launched from */etc/inittab* or not. But this requires the user to set the flag correctly during both automatic and interactive invocations. A common error would be for a user to examine the launching command in */etc/inittab* and then use it verbatim interactively.

Regardless of what approach is used, all the above mentioned pitfalls must still be recognized and avoided.

In the final analysis it seems that launching daemons from */etc/inittab*, as opposed to */etc/rc*, is unnecessary for the following reasons: (1) Relying on `init` to respawn a daemon is really masking a bug in the daemon; the daemon should never terminate by itself.† (2) Changing run states is an unusual occurrence on most systems; usually a system will move to multi-user mode and stay there.

Conclusions

Without following the above rules, strange symptoms which are hard to track down often result. Many times the errant daemon program is the last thing suspected (e.g., when terminal settings are not reset after logout). Other times it is the daemon that silently and mysteriously dies (e.g., when it attempts background I/O on a job control system). Frequently these symptoms only begin occurring well after the "debug period" for the daemon.

Example

The example below collects the above coding fragments into a single routine which a daemon calls to detach itself from the context of a login session.

```
/* Detach a daemon process from login session context.
 *
 * (This is a skeleton; add error condition checking and handling.)
 */
#include <signal.h>
#include <stdio.h>
#ifdef BSD
#include <sys/file.h>
#include <sys/ioctl.h>
#endif
sessdetach()
{
    int fd;          /* file descriptor */

    /* If launched by init (process 1), there's no need to detach.
     * Note: this test is unreliable due to an unavoidable race
     */
}
```

† One interesting counterexample is that some systems (e.g., ACSnet) allow system administrators to reset things by killing the appropriate daemons. It's nice to have the daemon start correctly (i.e., right arguments) by itself through the auspices of */etc/inittab*. However, it's arguably better to have the daemon catch the termination signal and perform the reset without actually terminating; this may even be essential in the case of orderly shutdown of operations such as line printer spooling.

;login:

```
    * condition if the process is orphaned.
    */
    if (getppid() == 1)
        goto out;
    /* Ignore terminal stop signals */
#ifdef SIGTTOU
    signal(SIGTTOU, SIG_IGN);
#endif
#ifdef SIGTTIN
    signal(SIGTTIN, SIG_IGN);
#endif
#ifdef SIGTSTP
    signal(SIGTSTP, SIG_IGN);
#endif
    /* Allow parent shell to continue.
    * Ensure the process is not a process group leader.
    */
    if (fork() != 0)
        exit(0);    /* parent */
    /* child */
    /* Disassociate from controlling terminal and process group.
    *
    * Ensure the process can't reacquire a new controlling terminal.
    * This is done differently on BSD vs. AT&T:
    *
    * BSD won't assign a new controlling terminal
    * because process group is non-zero.
    *
    * AT&T won't assign a new controlling terminal
    * because process is not a process group leader.
    * (Must not do a subsequent setpgrp(!))
    */
#ifdef BSD
    setpgrp(0, getpid());    /* change process group */
    if ((fd = open("/dev/tty", O_RDWR)) >= 0) {
        ioctl(fd, TIOCNOTTY, 0); /* lose controlling terminal */
        close(fd);
    }
#else /* AT&T */
    setpgrp(); /* lose controlling terminal & change process group */
    signal(SIGHUP, SIG_IGN); /* immune from pgrp leader death */
    if (fork() != 0) /* become non-pgrp-leader */
        exit(0);    /* first child */
    /* second child */
#endif
out:
    for (fd = 0; fd < _NFILE; fd++)
        close(fd); /* close all file descriptors */
    chdir("/"); /* move current directory off mounted filesystem */
    umask(0); /* clear any inherited file mode creation mask */
    return;
}
```

Nye »hak«-sekvenser til system V

```

+      Start understregning.
+#     Nyt:          # = 0 normal understregning
                    # = 1 dobbelt understregning
                    # = 2 punkteret understregning
                    # = 3 punkteret, dobbelt understregning

~#    Nyt: Skærmbkontrol # = 0 tænd skærm
                    # = 1 sluk skærm ( »crt save« )

E      Sideskift
E#     Nyt          # = 0 sideskift
                    # = 1 skift til bakke 1
                    # = n skift til bakke n
                    # > 0 genererer ikke sideskift, hvis muligt

F#     Side længde #      sidelængde i linjer
F#;$   Nyt         #      sidelængde i linjer
                    $      topmargen

G      Dobbelt bredde skrift
G#     Nyt          # = 0 dobbelt bredde.
                    # = 1 dobbelt højde
                    # = 2 dobbelt bredde, dobbelt højde

i      Invers
i#     Nyt          # = 0 normal invers
                    # = 1 skrift på grå baggrund

y#    Nyt: Skrift type # = 0 den skrift printereren skriver hurtigst
                    # = 1 den skrift printereren skriver »pæneste«
                    # > 1 andre

W#    Nyt: Vælg visning af vindue no #

g#;$   Definer scroll region, første linje i regionen er #
                    første linje efter regionen er $
Nyt:   # = 0 første linje er den, markøren står i
Nyt:   $ = 0 sidste linje er den, markøren står i

`      Genskab CPI, LPI, LM og set »vognen« ved ny venstre margen.
`#     Nyt          # = 0 som ` uden argumenter.
                    # = 1 Genskab CPI uden at flytte »vognen«.

```

Korte »hak« sekvenser for linjegråfik

```

À som T0;1   È som T8;1  Ṛ   à som T16;1   è som T24;1  Ṛ
Á som T1;1   É som T9;1  Ṛ   á som T17;1   é som T25;1  Ṛ
Â som T2;1   Ê som T10;1 |   â som T18;1   ê som T26;1  Ṛ
Ã som T3;1   Ë som T11;1 -   ã som T19;1   ë som T27;1  -
Ä som T4;1   Ì som T12;1 ˘   ä som T20;1   ì som T28;1 ˘
Å som T5;1   Í som T13;1 ˘   å som T21;1   í som T29;1 ˘
Æ som T6;1   Î som T14;1 ˘   æ som T22;1   î som T30;1 ˘
Ç som T7;1   Ï som T15;1 ˘   ç som T23;1   ï som T31;1 ˘

```

Nyt: Det er tilladt med »hak« sekvenser inde i »hak« sekvenser.

VOVVOV

Nummer 2

23. september 1987

I dette nummer:

Faldgruber med associerede printere
Advarsel vedrørende hardware handshake
Uucp-nettet
Netnews
System V Release 3.1
Opfordring

1. Faldgruber med associerede printere

Hovedregel: Under print-out kan terminalen ikke benyttes normalt. Så er det nemlig printeren, der bestemmer farten, og hvis den melder busy, stopper output til terminalen. Der vil i bedste fald være skumgummi i tasterne.

Så vidt jeg ved, er DDE ene på verdensmarkedet med generel support af associerede printere. At det er en kompliceret sag, kan forhåbentligvis anes ud fra følgende korte beskrivelse.

Følgende forhold skal være i orden:

Handshake skal laves omhyggeligt. Ved overgang til System V bliver mange skærme omkonfigureret. Den typiske handshake vil være xon/xoff mellem Supermax og terminal (her indgår ofte et modem) samt hardwarehandshake mellem terminal og printer (begrænsning i printeren og/eller terminalen). Det er nødvendigt, at terminalen kan omsætte printerens busy-signal til xon/xoff tegn mod Supermaxen.

Ingen tegnfølge til printeren må falde sammen med skærmens »print-off« sekvens. Den pågældende sekvens må undværes eller omskrives i VTI-tabellen for printeren.

Alle tegn til printeren skal videresendes af terminalen. Vi har f.eks. Wyse95, der »sluger« null-tegn. Dette tegn benyttes af mange printere under downloadning af specielle tegnmatricer til nationale tegn.

Denne type terminalfejl kræver omskrivning/nedgradering af printerens VTI-tabel, så problematiske tegn undgås.

Terminalen skal have tid til at fordøje sine egne funktioner. Vi har f.eks. Ampex230, der somme tider »glemmer« at videresende et tegn til printeren, fordi den har travlt med et eller andet andet, der kan være tasttryk, scroll eller lign.. Hvis fejlen opstår ved tasttryk, kan den ikke rettes! I dette tilfælde må brugeren informeres om at holde fingrene fra tastaturet, mens der skrives til printeren. Hvis fejlen opstår i forbindelse med terminalsekvenser, kræves der en omskrivning af terminalens VTI-tabel med fyldtegn efter generende sekvenser. Det kan være meget svært at indkredse de generende sekvenser, og der må i almindelighed et eller to dataskoper til hjælp. Det kan tænkes, at visse sådanne finpudsninger er gået tabt ved overgang fra System III til System V.

Sekvenser må ikke »skæres over«. Hvis f.eks. 'esc' '*' til terminalen skæres over af et 'p' til printeren, ser den samlede tegnfølge ud som følger:

'esc' 'esc' 'printon' 'p' 'esc' 'printoff' '*'

Dette kan medføre, at sekvenser går tabt, eller at output kommer forkert ud på den gale enhed.

SIOC, DIOC2 og SIOC2 garanterer, at enhver VTI-funktion kan ekspanderes til 176 tegn uden risiko for at blive skåret over. NIOC garanterer en ekspandering til 128

tegn. Eksempler på store ekspansioner ses ofte ved semigrafik. Færdige programskrevne sekvenser vil normalt kun kunne skæres over, hvis de sendes som enkelttegn, eller hvis der taster på tastaturet under output til printeren.

Der arbejdes på at teste konkrete terminal/printer-par til normalt brug, og der vil senere blive offentliggjort en liste over validerede kombinationer med tilhørende specialversioner af VTI-tabeller.

es [BP]

2. Advarsel vedrørende hardware handshake

NTC og Supermax signallerer busy på to forskellige ledninger!

Det har kun betydning ved store mængder hurtigt input til Supermax.

Normalt tilsluttes ydre enheder ved hjælp af et 8804/8805 kabel, hvor konektor 1 tilsluttes Supermax eller NTC, mens konektor 2 tilsluttes den ydre enhed. Supermax bremser inputstrømmen med DTR (ben 20), der er forbundet til den ydre enheds DSR (ben 6). NTC bremser inputstrømmen med RTS (ben 4), der er forbundet til den ydre enheds CTS (ben 5).

8804/8805 kablet er altså kun korrekt sammen med en Supermax, hvis den ydre enhed »smager« på DSR. 8804/8805 kablet er endvidere kun korrekt sammen med en NTC, hvis den ydre enhed »smager« på CTS.

Ovenstående er affødt af problemer med hardwarehandshake i forbindelse med filoverførsler fra PC-er.

es [BP]

3. Uucp-nettet

Uucp-nettet inden for DDE er blevet og bliver i disse dage udvidet til også at omfatte maskinerne *monalisa*, *mother*, og *scheuer-case*. Dette betyder at ansatte i ADM, IS, US, OS, og LISDA er kommet eller meget snart kommer på uucp-mail-systemet.

Dermed er uucp-nettet inden for DDE udvidet til at omfatte de allerfleste i DDE-Herlev.

ct [BP]

4. Netnews

Som lovet i sidste nummer af Vovov er der sket fremskridt med hensyn til læsning af uucp-netnews i DDE. Jeg skal derfor nedenfor gøre nærmere rede for hvad netnews er for noget, og hvorledes det bruges.

4.1 Hvad er netnews?

Netnews er en verdensomspændende »opslagstavle«. Den distribueres over uucp-nettet, og man har via netnews mulighed for at få kildetekster til diverse programmer, deltagelse i diskussioner om UNIX, C, ADA, græsk kultur, madopskrifter, kommunikationsprotokoller, m.m. Netnews er opdelt i nyhedsgrupper, der hver især er dedikeret til et specielt emneområde.

4.2 Hvordan kommer netnews hertil?

Netnews sendes hertil hver morgen fra *diku* (1-2 megabytes dagligt!). Det havner så på *dde*-maskinen (vores centrale uucp-netmaskine), hvorfra passende portioner så bliver overført til andre maskiner i DDE. Da det altså drejer sig om store datamængder, og da pladsen på forskellige maskiner kan være kneben, vil det blive nødvendigt med tilsagn fra jer om, hvilke nyhedsgrupper I vil have overført fra *dde*-maskinen til jeres maskine.

4.3 Hvordan læser jeg netnews?

Blandt de nyhedsgrupper der overføres til én maskine kan man »abonnere« på visse bestemte nyhedsgrupper. Dette sker ved at man i sit *home-directory* opretter en fil ved navn *.newsrc* med fx følgende indhold:

```
options -n comp.lang.c,soc.culture.greek
```

hvor man lister alle de nyhedsgrupper man vil abonnere på.

Når man logger på, køres i */etc/profile* et program ved navn *checknews*. Hvis *checknews* konstaterer at der er kommet nye artikler i en nyhedsgruppe man abonnerer på, udskrives meddelelsen »There is news« på skærmen.

Checknews kan naturligvis også køres manuelt (manualside er vedlagt).

Netnews læses med programmet *vnews* (manualside er vedlagt).

4.4 Hvordan sender jeg netnews?

Man benytter programmet *postnews* (manualside er vedlagt). Alternativt kan man sende »follow-up« artikler medens man er i *vnews*.

En af de ting man skal tage stilling til, når man sender netnews, er hvor vid udbredelse éns artikel skal have. Man kan bl.a. vælge mellem

- local Inden for DDE
- dk Danmark
- eunet Europa
- world Hele verden

Test af *postnews* kan foregå ved at sende data til nyhedsgruppen *dde.test*, som har lokal udbredelse, uanset hvad du skriver.

Husk at dine artikler bliver sendt viden om og bliver læst af mange mennesker, så du bør være omhyggelig og forsigtig med ordvalg — netnews er meget mere end den lokale opslagstavle i Brugsen.

Visse nyhedsgrupper er »modereret«, altså censureret. Dette betyder at man ikke kan sende direkte til disse. I stedet vil éns indlæg automatisk blive mailed til en person som sørger for indlægningen.

4.5 Hvilke nyhedsgrupper findes?

En oversigt over de i Danmark forekommende nyhedsgrupper er vedlagt.

4.6 Hvad skal jeg gøre nu?

Find ud af hvilke nyhedsgrupper I vil have fremført til jeres maskine, og meddel det til mig (Claus Tøndering), så skal jeg sørge for det videre fornødne.

4.7 Kan jeg kigge i nyhedsgrupper, der ikke findes på min maskine?

Ja. I sidste nummer af *Vovvov*, afsnit 4, blev det beskrevet hvorledes man kunne hente *netnews* ved at sende post til *archive@diku*. Denne archive funktion er nu lagt ind på *dde*, så man i stedet for at sende til *archive@diku* kan sende til *archive@dde*, hvilket burde give en hurtigere service.

4.8 Andre spørgsmål

Det ovenstående har kun været en kortfattet beskrivelse, hvis du har spørgsmål, er du velkommen til at henvende dig til mig. Jeg ligger også inde med stakkevis af yderligere dokumentation.

ct [BP]

5. System V Release 3.1

Vort hidtidige System V styresystem har haft betegnelsen »Release 2.1«. Som det vil være nogle bekendt, arbejdes der for tiden på indførelse af en ny release med nummeret 3.1. De vigtigste nyskabelser i denne release er følgende:

5.1 Streams

En ny mekanisme i styresystemet går under navnet STREAMS-mekanismen. Der er tale om en ny metode til at indlægge bl.a. kommunikationsprogrammer i systemet. Den almindelige bruger vil næppe mærke noget til dette.

5.2 RFS

RFS, *Remote File Sharing*, er et distribueret filsystem. Hvis to datamaskiner er forbundet via et netværk med en passende kommunikationsprotokol kan de dele filer. Det fungerer — kort fortalt — på den måde at den ene datamat offentliggør dele af sit filhierarki. Den anden datamat kan så få adgang til disse offentliggjorte filer som om de var en del af den eget filhierarki.

Meningen er at alle filoperationer der kan udføres lokalt også skal kunne udføres over det distribuerede filsystem. Dette gælder dog ikke for de filoperationer der ikke er en del af SVID, for disse understøttes ikke nødvendigvis af andre maskiner på netværket.

5.3 8-bits utilities

En lang række programmer er omlagt til 8-bits udgaver, så de kan håndtere nationale tegn. Det gælder fx *vi*, *sh*, *grep*, og *yacc*.

5.4 Ny konfigureringsmekanisme

Styresystemet vokser og vokser. Der bliver derfor behov for en mere fleksibel konfigureringsmekanisme, som giver systemadministratorer mulighed for at

bestemme hvilke dele af styresystemet man ønsker på sin maskine.

Hvis man ønsker at bruge streams og RFS, er man nødt til at bruge Winchester-boot på sin maskine.

5.5 Kompatibilitet

Alle programmer der kan køre under release 2.1 kan køre uændret under release 3.1.

ct [BP]

6. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes – brug fx en diskette, eller send mig artiklen med uucp-mail (postadresse *ct*).

ct [BP]

NAME

checknews - check to see if user has news

SYNOPSIS

checknews [ynqevvN] [newsgroup list] [readnews options]

DESCRIPTION

Checknews reports whether or not the user has news.

- y reports "There is news" if the user has news to read. If the -N flag is given, then the newsgroups requested are also printed.
- n reports "No news" if there isn't any news to read.
- q causes *checknews* to be quiet. Instead of printing a message, the exit status indicates news. A status of 0 means no news, 1 means there is news.
- v alters the -y message to show the name of the first newsgroup containing unread news. Doubling v (e.g. -vv) will cause an explanation of any claim of new news, and is useful if *checknews* and *readnews*(1) disagree on whether there is news.
- e executes *readnews* if there is news.
- N causes the next argument to be read and interpreted as a comma-separated list of newsgroups to be checked.

If there are no options, -y is the default.

FILES

~/.newsrc	Active newsgroups
/usr/lib/news/active	Options and list of previously read articles

SEE ALSO

inews(8), postnews(1), readnews(1), vnews(1), news(5), newsrc(5), expire(8), recnews(8), sendnews(8), uurec(8)

BUGS

The -N flag should really be named -n to be consistent with other news programs, but -n was already used. If the -v flag is used with the -N flag, the first newsgroup in the list where there is news should be printed instead of the entire list. If the -N flag is used and *readnews* is invoked (with -e) it does not restrict news reading to those groups checked, but reads all newsgroups where there is new news.

NAME

postnews - submit news articles

SYNOPSIS

postnews [article]

DESCRIPTION

Postnews is a program that calls *inews*(8) to submit news articles to USENET. The commands should be self-explanatory, however you may type "?" to most prompts to get a list of the possible options (except for the "Keywords" of the article, etc). It will prompt the user for the title of the article (which should be a phrase suggesting the subject, so that persons reading the news can tell if they are interested in the article), for the newsgroup, and for the distribution.

The distribution is typically a geographic region or corporate region. Typing "?" will get you a list of the possible distributions. You should use the minimum distribution that will serve your purpose for posting the article. For example, if you are selling your car in New Jersey, it is doubtful that someone in California (or Europe) would be willing to buy it. If you don't restrict the distribution to your local area, you will cause this article to be transmitted unnecessarily around the world. Currently, with a distribution of **world**, the article will be seen in the United States, Canada, Europe, Japan, Korea and other places. A distribution header will, if given, be included in the headers of the article, affecting where the article is distributed to.

After entering the title, newsgroup, and distribution, the user will be placed in an editor. If **EDITOR** is set in the environment, that editor will be used. Otherwise, *postnews* defaults to *vi*(1).

An initial set of headers containing the subject and newsgroups will be placed in the editor, followed by a blank line. The article should be appended to the buffer, after the blank line. The initial headers can be changed, or additional headers added, while in the editor, if desired.

After you have finished typing in your article, you have the option of sending it, listing it, quitting without sending it or saving it in a file so you can finish editing it and post it later.

For posting news from a program, see *inews*(8).

If the file */usr/lib/news/recording* is present, it is taken as a list of "recordings" to be shown to users posting news. (This is named after the recording you hear when you dial "information" in some parts of the U.S., asking you to stop and think if you really want to do this, but not actually preventing you.) The recording file contains lines of the form:

newsgroup-specifier TAB filename

for example:

comp.all comp.recording
local.all,!local.test local.recording

Any user posting an article to a newsgroup matching the pattern on the left will be shown the contents of the file on the right. The file is found in the **LIB** directory (often */usr/lib/news*). The user is then told to hit DEL to abort or RETURN to proceed. The intent of this feature is to help companies keep proprietary information from accidentally leaking out.

SEE ALSO

Mail(1), binmail(1), checknews(1), inews(8), mailx(1), readnews(1), vi(1), news(5), newsrc(5), expire(8), recnews(8), sendnews(8), uuexec(8)

NAME

vnews - read news articles

SYNOPSIS

vnews [**-a** *date*] [**-n** *newsgroups*] [**-t** *titles*] [**-rxuc**]

vnews **-s**

vnews **-K**

DESCRIPTION

Vnews is a program for reading USENET news. It is based on *readnews*(1) but has a CRT oriented interface. The list of available commands is quite similar, although since *vnews* is a "visual" interface, most *vnews* commands do not have to be terminated by a newline.

Vnews uses all but the last two lines of the screen to display the current article. The next-to-last line is the secondary prompt line, and is used to input string arguments to commands. The last line contains several fields. The first field is the prompt field. If *vnews* is at the end of an article, the prompt is "next?"; otherwise the prompt is "more?". The second field is the newsgroup field, which displays the current newsgroup, the number of the current article, and the number of the last article in the newsgroup. The third field contains the current time, and the last field contains the word "mail" if you have mail. When you receive new mail, the bell on the terminal is rung and the word "MAIL" appears in capital letters for 30 seconds.

Vnews without any arguments prints unread articles.

The following flags determine the selection of articles.

-a [*date*]

Select articles posted after the given *date* (in *getdate*(3) format).

-n *newsgroups*

Select articles belonging to *newsgroups*.

-t *titles* Select articles whose titles contain one of the strings specified by *titles*.

-r Print the articles in reverse order.

-x Ignore *.newsrc* file. That is, select articles that have already been read as well as new ones.

-u Update the *.newsrc* file every 5 minutes, as in the case of an unreliable system. (Note that if the *.newsrc* file is updated, the **x** command will not restore it to its original contents.)

If the **-c** flag is specified, *vnews* will print the first page of the article, instead of just the header.

You can use the **-s** flag to print the newsgroup subscription list.

If you haven't read news in a while (or if you have never read news!) you can do *vnews -K* to Kill (mark as read) all of the articles in the groups to which you are subscribed.

Vnews maintains a *.newsrc* file in the your home directory that specifies all news articles already read. It is updated at the end of each reading session unless the **-x** option was specified. If the environment variable **NEWSRC** is present, it should be the path name of a file to be used in place of *.newsrc*.

If you wish, an options line may be placed in your *.newsrc* file. This line starts with the word **options** (left justified) followed by the list of standard options just as they would be typed on the command line. Such a list may include: the **-n** flag along with a newsgroup list and/or the **-r** or **-t** flag. Continuation lines begin with a space or tab character.

ENVIRONMENT

Options can be specified in the **NEWSOPTS** environment parameter. Where conflicts exist, options on the command line take precedence, followed by the **.newsrc options** line, and lastly the **NEWSOPTS** parameter.

When the user uses the direct reply command, the environment parameter **MAILER** will be used to determine which mailer to use. The default is usually */bin/mail*.

If **EDITOR** is set, it will be used in place of the default editor on your system to edit replies and follow-ups.

If **NAME** is set, it will be used as your full name when posting news or submitting a follow-up. If it is not set, the name will be taken from the file *.name* in your home directory. If this file is not present, the name will be taken from */etc/passwd*.

If **NEWSARCHIVE** is set, a copy of any articles you post or follow-up to, will be saved in the specified file. If it is the null string, they will be copied in *author_copy* in your home directory.

If **NEWSBOX** is set, the filename you specify when you save or write a file will be prepended with **NEWSBOX** unless the filename is an absolute pathname.

If **NEWSRC** is set, it will be used in place of the *.newsrc* file in your home directory.

If **ORGANIZATION** is set, it will be used as the name of your organization whenever you post an article. The default is compiled in and is usually correct. Typically, you would only use this if you were reading news at a site other than normal. (Or if you are trying to be cute.)

COMMANDS

Each *vnews* command may be preceded by a count. Some commands use the count; others ignore it. If count is omitted, it defaults to one. Some commands prompt for an argument on the second line from the bottom of the screen. Standard UNIX erase and kill processing is done on this argument. The argument is terminated by a return. An interrupt (DEL or BREAK) gets you out of any partially entered command.

In the following table, ^B is used as a shorthand for Control-B.

Command	Meaning
CR	A carriage return prints more of the current article, or goes on to the next article if you are at the end of the current article. A SPACE is equivalent to CR .
^B	Go backwards <i>count</i> pages.
^F	Go forward <i>count</i> pages.
^D	Go forwards half a page.
^U	Go backwards half a page.
^Z	Go forwards <i>count</i> lines.
^E	Go backwards <i>count</i> lines.
^L	Redraw the screen. ^L may be typed at any time.
b	Back up one article in the current group.
c	Cancel the article. Only the author of the article or the super user can do this.
e	Erase. Forget that this article was read.
f	Submit a follow-up article. You will be placed in your EDITOR to compose the text of the follow-up.

- h** Go back to the top of the article and display only the header.
- l** Redisplay the article after you have sent a follow-up or reply.
- m** Move on to the next item in a digest.
- n** No. Go on to the next article without printing current one. `.` is equivalent to `n`. This is convenient if your terminal has a keypad.
- p** Show the parent article (the article that the current article is a follow-up to). This doesn't work if the current article was posted by A-news or notesfiles. To switch between the current and parent articles, use the `-` command. Unfortunately, if you use several `p` commands to trace the discussion back further, there is no command to return to the original level.
- q** Quit. The `.newsrc` file will be updated unless `-x` was on the command line.
- r** Reply. Reply to article's author via mail. You are placed in your **EDITOR** with a header specifying "To", "Subject", and "References" lines taken from the message. You may change or add headers, as appropriate. Add the text of the reply after the blank line, and then exit the editor. The resulting message is mailed to the author of the article.
- R** This is the same as `r` except the body of the article is included in your mail message for you.
- ESC-r** Reply directly. You are placed in your **MAILER** as if you had run it specifying the author of the article as the recipient of a letter.
- s** [*file*] Save. The article is appended to the named file. The default is *Articles*. If the first character of the file name is "`|`", the rest of the file name is taken as the name of a program, which is executed with the text of the article as standard input. If the first character of the file name is "`/`", it is taken as the full pathname of a file. If **NEWSBOX** (in the environment) is set to a full pathname, and the file contains no "`/`", the file is saved in **NEWSBOX**. Otherwise, it is saved relative to **HOME**.
- ug** Unsubscribe to the current group. This is a two character command to ensure that it is not typed accidentally and to leave room for other types of unsubscribes (e.g. unsubscribe to discussion).
- v** Print the current version of the news software.
- w** Write. Like save `s`, except that the headers are not written out.
- x** Exit. Like quit except that `.newsrc` is not updated.
- y** Yes. Print the current article and go to the next.
- [n]A** Go to article number *n* in the current newsgroup.
- D** Decrypts a joke. It only handles rot 13 jokes. The **D** command is a toggle; typing another **D** re-encrypts the joke.
- H** Print a very verbose header, containing all known information about the article.
- K** Kill (mark as read) the rest of the articles in the current group. This is useful if you can't keep up with the volume in the newsgroup, but don't want to unsubscribe.
- N** [*newsgroup*]
Go to the next newsgroup or named newsgroup.
- [n]+** Skip *n* articles. The articles skipped are recorded as "unread" and will be offered to you again the next time you read news.

- Go back to last article. This is a toggle; typing it twice returns you to the original article.
- < Prompt for an article ID or the rest of a message ID. It will display the article if it exists.
- # Report the name and size of the newsgroup.
- ? Print a short help message.
- | Passes the rest of the command line to the shell. The environment variable **A** is set to the name of the file containing the current article. If the last character of the command is a **&**, then the **&** is deleted and the command is run in the background with **stdin**, **stdout** and **stderr** redirected to */dev/null*. If the command is missing, the shell is invoked. Use the **|** command (or essentially any other command) to turn on the display after the program terminates.

EXAMPLES

- vnews** Read all unread articles using the *visual* interface. The *.newsrc* file is updated at the end of the session.
- vnews -n all !mod -r**
Read all unread articles except articles whose newsgroups begin with **mod**. in reverse order. The *.newsrc* file is updated at the end of the session.
- vnews -n all -a last thursday**
Print every unread article since last Thursday. The *.newsrc* file is updated at the end of the session.
- vnews -K**
Discard all unread news. This is useful after returning from a long trip.

FILES

<i>/usr/spool/news/newsgroup/number</i>	News articles
<i>/usr/lib/news/active</i>	Active newsgroups
<i>/usr/lib/news/vnews.help</i>	Help file for <i>visual</i> interface
<i>~/.newsrc</i>	Options and list of previously read articles

SEE ALSO

checknews(1), inews(8), postnews(1), readnews(1), vnews(1), getdate(3), news(5), newsrc(5), expire(8), recnews(8), sendnews(8), uurec(8)

NETNEWS NYHEDSGRUPPER

ct - 23. september 1987

Nedenfor er en komplet liste over alle de nyhedsgrupper vi har under *netnews* på dde-maskinen. I visse tilfælde har vi ikke præcise oplysninger om gruppernes formål. I så fald er beskrivelsen nedenfor erstattet af ???.

alt.gourmand	Recipies from the USENET Cookbook (troff & text).
alt.sources	Unmoderated group for posting sources (limited "reach"!).
comp.ai	Artificial intelligence discussions.
comp.ai.digest	Artificial Intelligence discussions. (Moderated)
comp.ai.edu	AI education.
comp.ai.neural-nets	???
comp.ai.nlang-know-rep	Natural language knowledge representation.
comp.ai.vision	Artificial vision.
comp.arch	Computer architecture.
comp.arch.parallel-sym	???
comp.binaries.amiga	Encoded public domain programs in binary. (Moderated)
comp.binaries.atari.st	Binary-only postings for the Atari ST. (Moderated)
comp.binaries.ibm.pc	Binary-only postings for IBM PC/MS-DOS. (Moderated)
comp.binaries.mac	Encoded Macintosh programs in binary. (Moderated)
comp.bugs.2bsd	Reports of UNIX* version 2BSD related bugs.
comp.bugs.4bsd	Reports of UNIX version 4BSD related bugs.
comp.bugs.4bsd.ucb-fixes	Bug reports/fixes for BSD Unix. (Moderated)
comp.bugs.misc	General bug reports and fixes (includes V7 & uucp).
comp.bugs.sys5	Reports of USG (System III, V, etc.) bugs.
comp.cog-eng	Cognitive engineering.
comp.compilers	Compiler construction, theory, etc. (Moderated)
comp.databases	Database and data management issues and theory.
comp.dcom.lans	Local area network hardware and software.
comp.dcom.lans.hyperchannel	???
comp.dcom.lans.v2lni	???
comp.dcom.modems	Data communications hardware and software.
comp.dcom.telecom	Telecommunications digest. (Moderated)
comp.doc	Archived public-domain documentation. (Moderated)
comp.doc.techreports	Lists of technical reports. (Moderated)
comp.editors	Editors.
comp.edu	Computer science education.
comp.edu.composition	???
comp.emacs	EMACS editors of different flavors.
comp.graphics	Computer graphics, art, animation, image processing.
comp.graphics.digest	Graphics software, hardware, theory, etc. (Moderated)
comp.hypercube	Massively parallel systems: hardware & software. (Moderated)
comp.lang.ada	Discussion about Ada*.
comp.lang.apl	Discussion about APL.
comp.lang.c	Discussion about C.
comp.lang.c + +	The object-oriented C + + language.
comp.lang.clu	???
comp.lang.forth	Discussion about Forth.
comp.lang.forth.mac	???
comp.lang.fortran	Discussion about FORTRAN.
comp.lang.icon	???
comp.lang.idl	???
comp.lang.lisp	Discussion about LISP.
comp.lang.lisp.franz	Discussion about Franz LISP.
comp.lang.lisp.x	???
comp.lang.misc	Different computer languages not specifically listed.
comp.lang.modula2	Discussion about Modula-2.
comp.lang.pascal	Discussion about Pascal.
comp.lang.postscript	???
comp.lang.prolog	Discussion about PROLOG.
comp.lang.rexx	???
comp.lang.scheme	???
comp.lang.smalltalk	Discussion about Smalltalk 80.
comp.lang.visual	???
comp.laser-printers	Laser printers, hardware & software. (Moderated)
comp.lsi	Large scale integrated circuits.
comp.lsi.cad	LSI CAD.
comp.mail.elm	Discussion and fixes for ELM mail system. (Moderated)

comp.mail.headers	Gatewayed from the ARPA header-people list.
comp.mail.mh	???
comp.mail.mhs	???
comp.mail.mhs.arpa	???
comp.mail.misc	General discussions about computer mail.
comp.mail.multi-media	???
comp.mail.uucp	Mail in the uucp network environment.
comp.misc	General topics about computers not covered elsewhere.
comp.music	???
comp.newprod	Announcements of new products of interest. (Moderated)
comp.org.decus	DEC* Users' Society newsgroup.
comp.org.fidonet	FidoNews digest, official news of FidoNet Assoc. (Moderated)
comp.org.usenix	USENIX Association events and announcements.
comp.os.aos	???
comp.os.cpm	Discussion about the CP/M operating system.
comp.os.cpm.amethyst	???
comp.os.eunice	The SRI Eunice system.
comp.os.minix	Discussion of Tanenbaum's MINIX system.
comp.os.misc	General OS-oriented discussion not carried elsewhere.
comp.os.os9	Discussions about the os9 operating system. (Moderated)
comp.os.research	Operating systems and related areas. (Moderated)
comp.os.rsts	???
comp.os.v	???
comp.os.vms	DEC's VAX* line of computers & VMS.
comp.os.xinu	The XINU operating system.
comp.periphs	Peripheral devices.
comp.periphs.printers	Printers.
comp.protocols.appletalk	Applebus hardware & software.
comp.protocols.ibm	IBM protocols.
comp.protocols.iso	ISO protocols.
comp.protocols.iso.dev-environ	???
comp.protocols.kermit	Info about the Kermit package. (Moderated)
comp.protocols.misc	Various forms and types of FTP protocol.
comp.protocols.pcnet	PCnet communication.
comp.protocols.pup	???
comp.protocols.tcp-ip	TCP and IP network protocols.
comp.protocols.tcp-ip.domains	???
comp.protocols.tcp-ip.ibmpc	TCP/IP for the IBM PC.
comp.risks	Risks to the public from computers & users. (Moderated)
comp.society	The impact of technology on society. (Moderated)
comp.society.futures	???
comp.software-eng	Software engineering.
comp.sources.amiga	Source code-only postings for the Amiga. (Moderated)
comp.sources.atari.st	Source code-only postings for the Atari ST. (Moderated)
comp.sources.bugs	For bug fixes and features discussion
comp.sources.d	For any discussion of source postings.
comp.sources.games	Postings of recreational software. (Moderated)
comp.sources.mac	Software for the Apple Macintosh. (Moderated)
comp.sources.misc	Posting of software. (Moderated)
comp.sources.unix	Postings of public-domain sources. (Moderated)
comp.sources.wanted	Requests for software and fixes.
comp.std.c	Discussion about C language standards. (Moderated)
comp.std.internat	Discussion about international standards.
comp.std.misc	Discussion about various standards. (Moderated)
comp.std.mumps	Discussion for the X11.1 committee on Mumps. (Moderated)
comp.std.unix	Discussion for the P1003 committee on UNIX. (Moderated)
comp.sys.amiga	Commodore Amiga: info&uses, but no programs.
comp.sys.apollo	Apollo computer systems.
comp.sys.apple	Discussion about Apple micros.
comp.sys.atari.8bit	Discussion about 8 bit Atari micros.
comp.sys.atari.st	Discussion about 16 bit Atari micros.
comp.sys.att	Discussions about AT&T microcomputers.
comp.sys.cbm	Discussion about Commodore micros.
comp.sys.cdc	Discussion about CDC systems.
comp.sys.celerity	Discussion about Celerity systems.
comp.sys.dec	Discussions about DEC computer systems.
comp.sys.dec.micro	Discussions about DEC micro computer systems.
comp.sys.encore	Discussion about Encore systems.
comp.sys.handhelds	???
comp.sys.hp	Discussion about Hewlett-Packard equipment.
comp.sys.ibm.pc	Discussion about IBM personal computers.
comp.sys.ibm.pc.digest	The IBM PC, PC-XT, and PC-AT. (Moderated)
comp.sys.ibm.pc.net	The IBM PC net.
comp.sys.ibm.pc.rt	The IBM PC RT.

comp.sys.intel	Discussions about Intel systems and parts.
comp.sys.intel.ipsc310	???
comp.sys.m6809	Discussion about 6809's.
comp.sys.m68k	Discussion about 68k's.
comp.sys.m68k.pc	Discussion about 68k-based PCs. (Moderated)
comp.sys.mac	Discussions about the Apple Macintosh & Lisa.
comp.sys.mac.digest	Apple Macintosh: info&uses, but no programs. (Moderated)
comp.sys.masscomp	The Masscomp line of computers. (Moderated)
comp.sys.misc	Discussion about computers of all kinds.
comp.sys.northstar	???
comp.sys.nsc.32k	National Semiconductor 32000 series chips.
comp.sys.proteon	???
comp.sys.pyramid	Pyramid 90x computers.
comp.sys.ridge	Ridge 32 computers and ROS.
comp.sys.sequent	Sequent systems, (esp. Balance 8000). (Moderated)
comp.sys.sgi	???
comp.sys.sun	Sun "workstation" computers. (Moderated)
comp.sys.super	???
comp.sys.tahoe	???
comp.sys.tandy	Discussion about TRS-80's.
comp.sys.ti	Discussion about Texas Instruments.
comp.sys.ti.explorer	???
comp.sys.transputer	Discussion about the transputer.
comp.sys.workstations	Various workstation-type computers. (Moderated)
comp.sys.xerox	Discussion about XEROX systems.
comp.sys.zenith	Discussion about Zenith systems.
comp.sys.zenith.z100	Discussion about the Zenith Z100 system.
comp.terminals	All sorts of terminals.
comp.terminals.bitgraph	???
comp.terminals.tty5620	???
comp.text	Text processing issues and methods.
comp.text.desktop	Technology & techniques of desktop publishing. (Moderated)
comp.theory	Computer theory.
comp.theory.cell-automata	???
comp.theory.dynamic-sys	???
comp.theory.finite-el	Finite element methods.
comp.theory.info-retrieval	Information retrieval systems. (Moderated)
comp.theory.self-org-sys	???
comp.unix	Discussion of UNIX* features and bugs. (Moderated)
comp.unix.cray	Discussion about Cray UNIX.
comp.unix.questions	UNIX neophytes group.
comp.unix.wizards	Discussions, bug reports, and fixes on and for UNIX.
comp.unix.xenix	Discussion about the Xenix OS.
comp.windows.misc	Various issues about windowing systems.
comp.windows.news	Sun Microsystems' News window system.
comp.windows.x	Discussion about the X Window System.
dde.test	DDE intern test.
diku.bugs	Bugs found in diku software
diku.crash	Reports about crash
diku.edbafd	Group for comp. dept.
diku.events	Events and announcements concerning diku
diku.games	Games on diku
diku.jobs	Job openings in Denmark or for people at diku
diku.jokes	Jokes
diku.micro	Group for micro systems
diku.micro.atari	For the ATARI micro
diku.micro.mac	For the Macintosh
diku.net	Group for networking
diku.net.lan	News about the BRIDGE lan on diku
diku.net.news	News about news
diku.net.uucp	Uucp net
diku.printers	Comments about the printers
diku.problems	General problems
diku.projects	Forum for discussing new projects, looking for interested
diku.software	Announcing about software
diku.software.local	Announcing locally developed software
diku.tapes	About tapes
diku.terminals	About terminals
diku.test	Test group for the newcommers etc.
diku.test.test	Moderated test group
dk	General news for Denmark
dk.general	General news for Denmark
dk.test	National wide test group
eunet.bugs.4bsd	Bug reports on 4.*bsd, limited (EUnet) "audience".

eunet.bugs.uucp	Bug reports on uucp, specifically EUUG uucp.
eunet.esprit	Discussion/info about Esprit projects.
eunet.followup	Followup group for eunet.general.
eunet.general	General discussions.
eunet.jokes	Jokes only Europeans can (do?) understand... :-)
eunet.micro.acorn	Discussion about Acorn micros.
eunet.news	Group for news topics, limited to EUnet.
eunet.news.group	Discussion on and proposals for new eunet newsgroups.
eunet.politics	(European) political discussions (and flames!).
eunet.sources	EUnet-wide (only!) group for posting sources.
eunet.test	EUnet-wide test group.
eunet.works	Workstations (specifically European) topics.
general	Articles that should be read by everyone on your local system
misc.jobs.misc	Discussion about employment, workplaces, careers.
misc.jobs.offered	Announcements of positions available.
misc.jobs.resumes	Postings of resumes and "situation wanted" articles.
misc.misc	Various discussions not fitting in any other group.
misc.psi	Paranormal abilities and experiences. (Moderated)
misc.security	Computer security. (Moderated)
misc.wanted	Requests for things that are needed (NOT software).
news.admin	Comments directed to news administrators.
news.announce.conferences	Calls for papers and conference announcements. (Moderated)
news.announce.important	General announcements of interest to all. (Moderated)
news.announce.newusers	Explanatory postings for new users. (Moderated)
news.config	Postings of system down times and interruptions.
news.groups	Discussions and lists of newsgroups.
news.lists	News-related statistics and lists. (Moderated)
news.misc	Discussions of USENET itself.
news.software.b	Discussion about B news software.
news.software.nntp	Discussion about NNTP news software.
news.software.notes	Notesfile software from the Univ. of Illinois.
news.stargate	Discussion about satellite transmission of news.
news.sysadmin	Comments directed to system administrators.
rec.arts.sf-lovers	Science fiction lovers' newsgroup.
rec.equestrian	Equestrianism.
rec.games.board	Discussion and hints on board games.
rec.games.bridge	Hobbyists interested in bridge.
rec.games.chess	Chess & computer chess.
rec.games.empire	Discussion and hints about Empire.
rec.games.frp	Discussion about Fantasy Role Playing games.
rec.games.go	Discussion about Go.
rec.games.hack	Discussion, hints, etc. about the Hack game.
rec.games.misc	Games and computer games.
rec.games.pbm	Discussion about Play by Mail games.
rec.games.programmer	Internet mailing list gatewayed into netnews.
rec.games.rogue	Discussion and hints about Rogue.
rec.games.trivia	Discussion about trivia.
rec.games.vectrex	Internet mailing list gatewayed into netnews.
rec.games.video	Discussion about video games.
rec.ham-radio	Amateur Radio practices, contests, events, rules, etc.
rec.ham-radio.packet	Discussion about packet radio setups.
rec.mag	Magazine summaries, tables of contents, etc.
rec.mag.fsfn	??? (Moderated)
rec.mag.otherrealms	Edited science fiction & fantasy "magazine". (Moderated)
rec.music.beatles	The Beatles.
rec.music.classical	Discussion about classical music.
rec.music.folk	Folks discussing folk music of various sorts.
rec.music.gaffa	Progressive music (e.g., Kate Bush). (Moderated)
rec.music.gdead	A group for (Grateful) Dead-heads.
rec.music.makers	For performers and their discussions.
rec.music.misc	Music lovers' group.
rec.music.synth	Synthesizers and computer music.
sci.astro	Astronomy discussions and information.
sci.bio	Biology and related sciences.
sci.bio.technology	Biological technology.
sci.crypt	Different methods of data en/decryption.
sci.electronics	Circuits, theory, electrons and discussions.
sci.lang	Natural languages, communication, etc.
sci.lang.japan	???
sci.math	Mathematical discussions and pursuits.
sci.math.num-analysis	Numerical analysis.
sci.math.stat	Statistics discussion.
sci.math.symbolic	Symbolic algebra discussion.
sci.med	Medicine and its related products and regulations.

sci.med.aids	Medicine subgroup for AIDS related topics. (Moderated)
sci.misc	Short-lived discussions on subjects in the sciences.
sci.philosophy	Discussion about philosophy.
sci.philosophy.meta	???
sci.philosophy.tech	Technical philosophy: math, science, logic, etc.
sci.physics	Physical laws, properties, etc.
sci.psychology	Internet mailing list gatewayed into netnews.
sci.research	Research methods, funding, ethics, and whatever.
sci.space	Space, space programs, space related research, etc.
sci.space.shuttle	The space shuttle and the STS program.
soc.culture.african	Discussions about Africa & things African.
soc.culture.celtic	Group about Celtics (*not* basketball).
soc.culture.esperanto	Esperanto.
soc.culture.greek	Group about Greeks.
soc.culture.indian	Group for discussion about India & things Indian.
soc.culture.japan	Japanese culture.
soc.culture.jewish	Group for discussion about Jewish culture & religion.
soc.culture.misc	Group for discussion about other cultures.
soc.human-nets	Computer aided communications digest. (Moderated)
soc.politics	Political problems, systems, solutions. (Moderated)
soc.politics.arms-d	Arms discussion digest. (Moderated)

VOVVOV

Nummer 3

12. oktober 1987

I dette nummer:

Efterlysning – eller – »Hovsa, pascaloversætteren gik ned!«
Nye Pascal'er på System V
UUCP-mail
Opfordring

1. Efterlysning – eller – »Hovsa, pascaloversæt- teren gik ned!«

Tja – det hænder jo for de fleste programmer, at de fejler, og pascaloversætterne er ingen undtagelse. Især har de en vis evne til at »gå ned« efter at have opdaget specielle typer syntaksfejl – ofte følgefejl på <identifiers>, der ikke kendes på det tidspunkt, de anvendes. Med andre ord: oversætterne er ikke særlig fejltolerante.

Grunden til, at oversætterne fejler i disse situationer, er, at kildeteksten til dem (som er skrevet i pascal) stadig indeholder forsøg på at referere til data igennem NIL-pointere.

Det vil være en stor hjælp for mig i mit arbejde med at finde disse NIL-referencer, hvis I, når oversætteren fejler, gider sende mig en diskette indeholdende den kildetekst, der var under oversættelse på pågældende tidspunkt. Jeg behøver kun de pascalfiler, der indgår i pågældende oversættelse, da jeg jo ikke skal linke programmerne. I kan også sende det over UUCP-nettet, hvor min postadresse er *abc*.

På forhånd tak!

abc [BP]

2. Nye Pascal'er på System V

Nu skulle de nye versioner af fortolket pascal og pascal assembler være på vej ud til afdelingerne. Samtidigt er System V Pascal Manualen omsider blevet færdiggjort. For at

I ikke skal bruge al jeres tid på at læse manualen, følger her en kort liste over de ændringer, der er relevante, med angivelse af hvilke afsnit i manualen, der beskriver dem.

En ændring, der kan give anledning til mærkelige syntaks- og linkerfejl, er, at vi med denne version af Pascal går over til at genkende navne med 14 betydende tegn i stedet for som hidtil 8. Det betyder, at gamle programmer, hvor der er trykfejl i navne efter det ottende tegn, nu vil give syntaksfejl. Samtidigt skal man sørge for, at eventuelle egne eksterne C-rutiner nu hedder det samme i C-erklæringen og pascal-erklæringen til og med 14. tegn.

Pascal User's Guide

Afsnit 2.5.3:

ny repetitionssætning – LOOP-EXITIF-ENDLOOP.

Afsnit 2.7:

tekst-biblioteker (sprogafhængighed)

Afsnit 3.28:

LOADTEXT – sprogafhængighed

Afsnit 3.63:

STDTEXT – sprogafhængighed

Appendix A:

læs det, selv om du tror, at du kender det!!!!

Running Pascal-Assembler Compiler

Afsnit 4:

lidt om stak- og heap-størrelser

Appendix B:

brugerhåndtering af run-time fejl

Udover ovennævnte tilføjelser/ændringer er der selvfølgelig rettet fejl hist og her. Hvis (når!) I konstaterer fejl i pascal'erne, så meld dem til mig, men helst skriftligt! Forslag til nye faciliteter og nye standard eksterne rutiner modtages også gerne.

abc [BP]

3. UUCP-mail

Det er nu ca. et halvt år siden jeg udsendte det første skrift om hvorledes UUCP-mail fungerer. I de seneste måneder er der kommet adskillige nye brugere på UUCP-nettet og det har vist sig at en repetition af hvorledes nettet fungerer er nødvendig. Dette afsnit vil derfor stort set være en gentagelse af hvad der tidligere er sagt, og læsere der føler sig helt fortrolige med UUCP-mail-systemet kan roligt springe dette afsnit over.

3.1 Hvad er UUCP-nettet?

UUCP-nettet er en (mere eller mindre) verdensomspændende sammenkobling af UNIX-datamater. At DDE er kommet på dette net betyder at vi nu har mulighed for at sende og modtage elektronisk post fra en lang række mennesker rundt om i verden. Desuden er en hel række maskiner inden for DDE koblet sammen, så vi kan sende elektronisk post til hinanden.

UUCP-mail-systemet må ikke forveksles med *netnews*-systemet som blev omtalt i sidste nummer af *Vovvov*. Ej heller må UUCP-mail-systemet forveksles med kontorsystemets postfunktion. Derimod kan man godt benytte UUCP-mail-systemet fra Uniplex (men detaljerne herom har jeg ikke check på).

3.2 Betingelser for at kunne bruge systemet

Følgende grundlæggende forhold skal være i orden for at man kan bruge mail-systemet:

1. Man skal have én fast maskine hvortil éns post bliver sendt.
2. På denne maskine skal man logge ind med et login-navn som er éns initialer i DDE.

3. Inden for DDE skal samtlige maskiner på UUCP-nettet vide på hvilken maskine man sidder. Hvis dette ikke er tilfældet, skal man henvende sig til Peter Heick og bede ham ordne det.

3.3 Brug af mail-systemet

Der findes to programmer til afsendelse og modtagelse af post. Det er *mail* og *mailx*. Af disse er *mailx* det nyttigste, idet det har flest faciliteter, men naturligvis er *mailx* så også vanskeligere at lære at bruge. Læs manualen!

3.4 Adresser

Når man sender skal man naturligvis kende adressen på den person man vil sende til.

Ved post inden for DDE er folks postadresse slet og ret deres initialer i firmaet. Hvis man fx vil sende post til mig (*ct*) kan man skrive:

`mailx ct`

(*Mailx* har nogle problemer med den salgsadresser — se afsnit 3.9.)

Da samtlige UUCP-maskiner i DDE ved hvor jeg sidder, vil posten automatisk blive sendt til min maskine, hvor jeg kan læse den.

Hvis man vil sende post ud af DDE over UUCP-nettet skal man kende modtagerens netværksadresse. Der findes to typer adresser: *!*-adresser og *@*-adresser.

En typisk *!*-adresse er:

`mcvax!ukc!uel!gw`

En typisk *@*-adresse er:

`gw@uel`

I de følgende to afsnit omtales disse adresseringsformer.

3.5 *!*-adresser

!-adresser bruges til at specificere den fulde vej fra computer til computer, som posten skal følge. Adressen `mcvax!ukc!uel!gw` betyder at personen *gw* sidder på maskinen *uel* som sidder på maskinen *ukc* som sidder på maskinen *mcvax* (som er den centrale UUCP-maskine i Europa — den står i Holland).

DDE er i UUCP-sammenhæng at betragte som én maskine ved navn *dde*. Denne »maskine« hænges på en maskine ved navn *diku* (som står på Datalogisk Institut ved

Københavns Universitet), som hænger på *mcvax*.

Hvis man derfor skal sende post til ovennævnte person, skal kommandoen fx være:

```
mailx diku!mcvax!ukc!uel!gw
```

Bemærk at *dde* ikke skal indgå i adressen.

Hvis du skal sende post til USA skal du muligvis vide, at *mcvax* kan snakke med en datamat i USA ved navn *uunet*.

Når jeg skal opgive min egen netværks-adresse over for andre i !-notation, bliver det altså *mcvax!diku!dde!ct*, eller eventuelt (til USA) *uunet!mcvax!diku!dde!ct*. Ofte angiver man at der skal sættes noget foran det indledende maskinnavn; så adressen vil typisk blive skrevet således: *...!mcvax!diku!dde!ct*.

Det er denne adresse der står på vores nye visitkort.

3.6 @-adresser

Det er jo besværligt at skulle kende hele postvejen fra maskine til maskine, og derfor findes der også en anden mulighed. Sagen er nemlig den at *diku* kender navnet på samtlige maskiner i UUCP-nettet. *diku* ved fx at *uel* hænger på *ukc* som hænger på *mcvax*. Det er derfor tilstrækkelige at skrive en adresse som følger:

```
mailx gw@uel
```

Når vores mailer ser en sådan adresse bliver den videresendt til *diku* som expanderer den efter behov. Og det er jo meget nemmere, ikke sandt? Derfor bør @-adresser som regel foretrækkes frem for !-adresser.

Ofte opgives @-adresse på formen: *gw@uel.uucp* eller *gw@uel.UUCP*. Dette afsluttende »uucp« eller »UUCP« er navnet på netværket, og kan udelades når man, som DDE, selv hænger på UUCP-netværket. Der findes imidlertid andre netværk (fx Arpa-nettet i USA), og når folk fra disse netværk skal sende post til UUCP-nettet skal »uucp« medtages

Når jeg skal opgive min egen netværks-adresse over for andre i @-notation, bliver det altså *ct@dde.uucp*. (Notationen med »UUCP« med store bogstaver bør ikke bruges.)

Man skal være opmærksom på at ikke alle

datamater forstår @-adresserne, hvorfor det ikke er alle som vi kunne sende post til os via vores @-adresse. Dette er baggrunden for at det er vores !-adresse og ikke vores @-adresse der står på de nye visitkort.

3.7 Hvordan skal jeg så angive min adresse?

Når man angiver sin adresse over for folk ude i den store verden er det klogt at angive begge adresser, og man bør altid tage det afsluttende »uucp« med i @-adressen.

Jeg plejer altid at underskrive mine elektroniske breve med mit navn, min stilling, firmaets navn, samt følgende:

E-mail: *...!mcvax!diku!dde!ct* or *ct@dde.uucp*

3.8 Et par advarsler

Husk, at det endnu er de færreste maskiner der kører med ISO-8859-tegnsættet. Man bør derfor i breve der går uden for DDE holde sig til ASCII-alfabetet, og man bør benytte { eller ae for æ, } eller aa for å, osv.

Der er ikke noget der hedder post-hemmelighed når det drejer sig om UUCP-nettet. Superbrugeren på alle maskiner som brevet passerer kan læse hvad skrevet. Pas derfor på ikke at sende fortrolige oplysninger uden for DDE, ligesom glødende kærlighedsbreve m.m. ikke bør forekomme, hvis man ikke ønsker pludselig at se en kopi på afdelingens opslagstavle.

3.9 Problemer med mailx

Mailx har problemer med at sende post inden for DDE. Hvis jeg vil sende post til *fha*, som ikke kører på samme maskine som jeg, vil *mailx* brokke sig til trods for at *mail* sagtens kan finde ud af det.

Dette problem kan løses på to måder. Man kan fx specificere adressen fuldt ud:

```
mailx fha@dde
```

selv om man sender post inden for DDE. En bedre løsning er imidlertid at få superbrugeren på ens maskine til at oprette en fil ved navn */usr/lib/mailx/mailx.rc* med følgende indhold:

```
set sendmail=/bin/mail
```

Dette får *mailx* til at undertrykke sin egen (forkerte) check på brugernavnet.

ct [BP]

4. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes – brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse ct).

ct [BP]

VOVVOV

Nummer 4

10. november 1987

I dette nummer:

System V og troff/nroff
Hvad sker der når strømmen går?
Nye newsgroups
Opfordring

1. System V og troff/nroff

En og anden har måske undret sig over at nroff/troff tekstformatteringsprogrammerne ikke er med på vores standarddistribution af System V. Det skyldes at AT&T ikke længere leverer nroff/troff med som en del af UNIX, men i stedet leverer et selvstændigt produkt kaldet *Documenters Workbench*. Dette produkt indeholder foruden nroff/troff også de tidligere kendte præprocessorer *eqn* og *tbl*. Desuden er der tilføjet endnu nogle præprocessorer nemlig *pic* og *grap*.

DDE har for øjeblikket en intern licens på *Documenters Workbench*, og den har været testet igennem et stykke tid i BP, og vist sig at være rimeligt velkørende. Ved henvendelse til BKN vil man kunne få en streamer-tape med *Documenters Workbench* samt dokumentation. En distributions licens kan købes for ca. 30.000 kr., og hvis behovet er der vil vi naturligvis gøre dette.

En yderligere fordel ved den troff der leveres med *Documenters Workbench*, er at den er device-uafhængig, hvilket vil sige at det ikke længere kun er den såkaldte C.A.T fotosætter der understøttes af troff. Troff (også kaldet ditroff) er med held anvendt til Canon LPB-A2 laserprinteren samt Chelgraphs RIP (Raster Image Processor, også anvendt af GMI). Navnlig med RIP opnås et ganske pænt resultat. (Dette blad er skrevet på en RIP.) I forbindelse med Canon laserprinterens kræves en proportional skriftkassette, hvor kun Garland skriften er supporteret for øjeblikket. Jeg mener at huske at der til den nye udgave af Canon

laserprinterens kan fås flere typer proportional skrifter, og disse vil ret let kunne tilføjes. For at få et indtryk af de forskellige præprocessorer findes vedhæftet nogle eksempler på *pic*, *grap*, *eqn* og *tbl*.

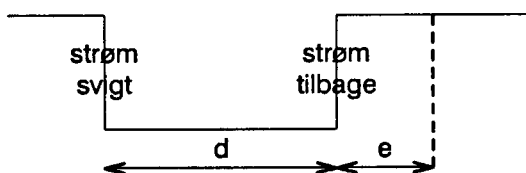
jk [BP]

2. Hvad sker der når strømmen går?

Alle har sikkert oplevet at netop som man sidder og har indtastet en masse programlinier går strømmen, og alt hvad man møjsommeligt har indtastet er hermed væk. Den oplevelse har DDE kunder tilsyneladende også haft, hvorfor man efterkom det ønske at udvikle en såkaldt UPS (Uninterruptable Power Supply). Princippet i denne er at hvis strømmen går sørger en elektronisk anordning straks for at skifte over til batteridrift. Hvis strømmen ikke kommer igen med det samme, bliver maskinen automatisk lukket ned.

Dette er i og for sig udemærket, men hvad har man så opnået set i forhold til situationen uden en UPS. Hvis strømmen går slukkes ens terminal, da denne ikke er tilkoblet UPS'en, altså har man ingen mulighed for at gemme noget som helst. Hvis strømmen kommer igen umiddelbart efter er der ikke sket alverden udover, at det skærbillede man stod i er væk. Kommer strømmen ikke igen med det samme bliver maskinen lukket automatisk ned og det er nøjagtigt lige så slemt som hvis man ingen UPS havde, da man ikke får mulighed for at

gemme data. Heldigvis er der mulighed for at programmer kan få oplysning om at strømmen er gået eller kommet igen, eller om at maskinen er ved at blive lukket ned. Essensen af alt dette er at hvis man skal have fordel af en UPS skal ens programmer også være kodet til at køre under en sådan. Endvidere er det sådan at de programændringer der skal foretages også vil være en fordel for programmer der kører på en maskine uden UPS. Det følgende er et forsøg på at klarlægge hvad der sker ved et strømsvigt og hvorledes man koder sine programmer så der mistes færrest data. Hændelesforløbet ved et strømsvigt kan skitseres således:



Her er afstanden d den tid maskinen er nede og afstanden e en tidskonstant styresystemet benytter inden den i visse situationer sender en bestemt type signal til alle processer.

Hvis tiden d er mindre end 30 sec. vil maskinen ikke blive lukket ned. Det der så sker er:

Hvis ens terminal er koblet som en fast terminallinie, hvilket er bestemt af `stty` flaget `clocal`, vil alle programmer modtage signalet `SIGPWR` efter tidsrummet e . Tidsrummet e er for øjeblikket 30 sec., hvilket skulle være tilstrækkeligt til at ens terminal har haft tid til at sunde sig oven på strømudfaldet.

Hvis ens terminallinie derimod er en modemlinie eller ens `clocal` flag ikke er sat, svarende til at flaget er `-clocal`, hvis man kører programmet `stty`, vil alle programmer der kører på terminalen modtage et `SIGHUP`, hvilket under normale omstændigheder bevirker at programmerne dør.

Hvis tidsrummet d er større end 30 sec. gælder selvfølgelig de ovennævnte betragtninger med hensyn til terminaler der er koblet som modemlinier, men yderligere vil systemet automatisk blive lukket ned via de indgange i `/etc/inittab` der har label

`powerfail`. Som vores system distribueres sker det via en shutdown med »grace« værdi 0, hvilket vil sige at systemet bringes ned hurtigst muligt. Under shutdown bliver programmet `killall` kørt. Dette program sender signalet `SIGTERM` til alle programmer, hvorefter det venter 30 sec. og sender signalet `SIGKILL` til de samme programmer, hvilket bevirker at programmerne med garanti dør. Programmerne har altså her 30 sec. til at gøre noget fornuftigt i inden de bliver fjernet.

Hvad gør man ved det?

Svaret er »signal handler«. Det vigtigste ved kodning af programmer der skal køre under en UPS, er håndtering af signaler. Både i C og Pascal er det muligt at opsætte »signal handler«, der fanger signaler og derefter udfører nogle passende handlinger inden programmet terminerer. Se i System V reference manualen under `signal(2)` for hvorledes man sætten en signal handler op.

Som et eksempel på et program der er kodet til at kunne fange en shutdown situation, og såmænd også et system crash, er editoren `vi`. Som noget af det første når `vi` startes op opsættes en signal handler for både `SIGHUP` og `SIGTERM`. Hvis et af disse signaler fanges vil `vi` »sync'e« sine temporære filer og kalde et program, der gemmer disse i et specielt directory (`/usr/preserve`), hvorefter der bliver sendt en meddelelse via UNIX mail, om at brugeren kan genskabe den fil vedkommende var igang med at rette da systemet gik ned. Endvidere har `vi` et program der kan køres når systemet bootes der kopierer temporære filer fra `/tmp` til `/usr/preserve` og sender mail til brugere der var ved at editere da systemet gik ned. Det `vi` ikke kan er at gentegne skærmbilledet ved signalet `SIGPWR`.

Med hensyn til vores NIOC, bør det bemærkes, at ved et strømsvigt af selv den korteste varighed mistes forbindelsen mellem NIOC on NTC, medmindre forbindelsen er permanent etableret har man altså ingen glæde af det `SIGPWR` der bliver sendt efter at strømmen er kommet tilbage.

Konklusionen må være at som et led i DDE's bestræbelser på at gøre vores systemer mere robuste overfor fejlsituationer, er en af

de ting der ligger først for, at få systemene til at kunne håndtere systemnedbrud og strømsvigt på en fornuftig måde, uden tab af data. Vejen er vist via adskillige standard UNIX utilities.

jk [BP]

3. Nye newsgroups

Der er fremkommet et fornuftigt ønske om at vi i *netnews* systemet opretter nogle newsgroups som er interne for DDE. I disse newsgroups kan man så diskutere problemer som vedrører forskellige DDE-produkter. Der er foreslået følgende newsgroups:

- dde.kalender
- dde.kartotek
- dde.kontor
- dde.oracle4
- dde.oracle5
- dde.post
- dde.regneark
- dde.sag
- dde.tekst
- dde.unix
- dde.vinduer

Jeg modtager meget gerne kommentarer og ønsker om ekstra DDE-newsgroups. I løbet af et par uger vil jeg så installere disse nye grupper.

ct [BP]

4. Opfordring

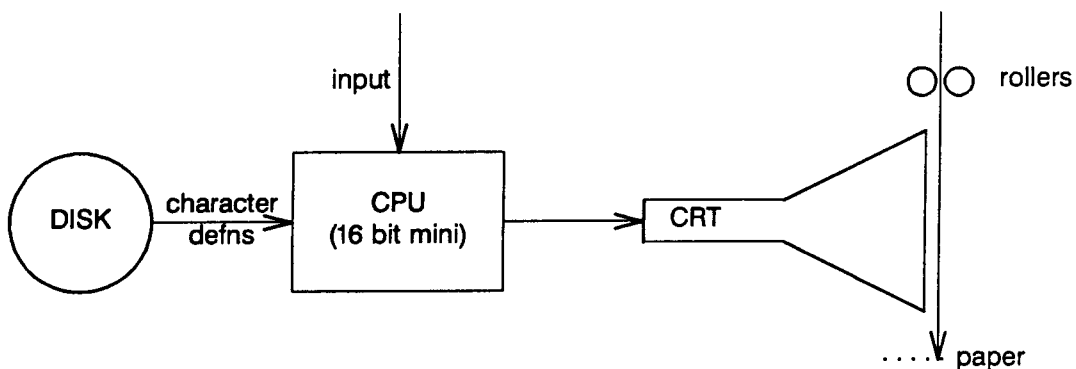
Vovvov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes — brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse *ct*).

ct [BP]

Eksempel på brug af pic:

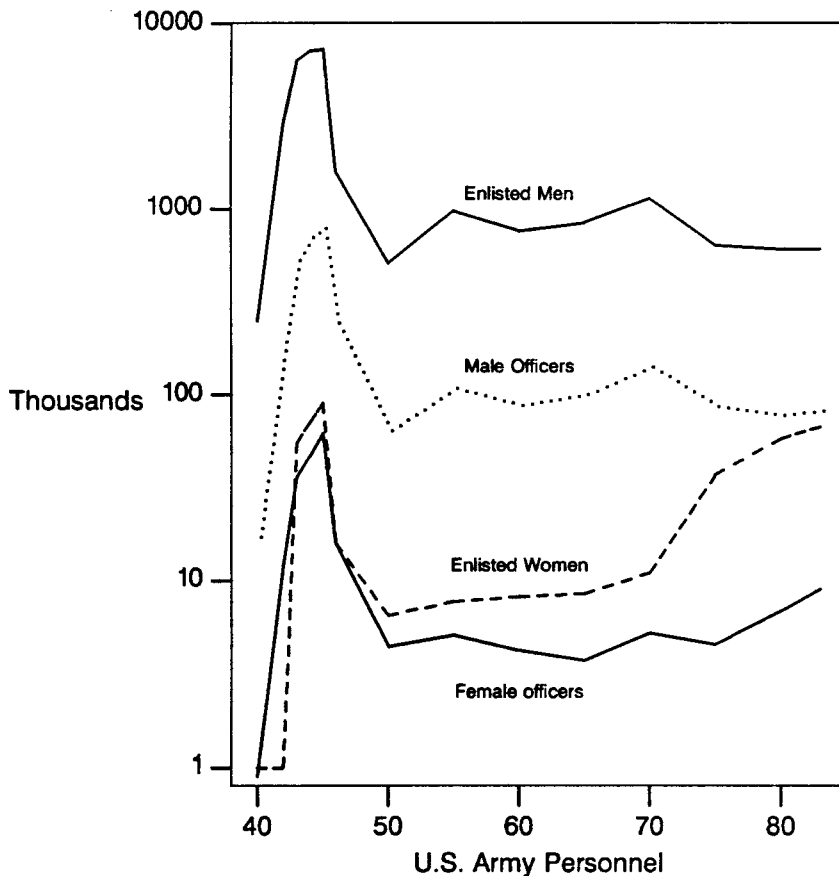
```
.PS 5
circle "DISK"
arrow "character" "defns"
box "CPU" "(16 bit mini)"
{ arrow <- from top of last box up "input " rjust }
arrow
CRT: " CRT" ljust
line from CRT - 0,0.075 up 0.15 \
then right 0.5 \
then righth 0.5 up 0.25 \
then down 0.5+0.15 \
then left 0.5 up 0.25 \
then left 0.5

Paper: CRT + 1.0+0.05,0
arrow from Paper + 0,0.075 to Paper - 0,0.5
{ move to start of last arrow down 0.025
  { move left 0.015; circle rad 0.05 }
  { move right 0.015; circle rad 0.05; " rollers" ljust }
}
" paper" ljust at end of last arrow right 0.25 up 0.25
line left 0.2 dotted
.PE
```



Eksempel på brug af grap:

```
.G1
frame ht 4 wid 3.2
coord x 38,85 y .8,10000 log y
label bot "U.S. Army Personnel"
label left "Thousands" left .3
draw of solid      # Officers Female
draw ef dashed     # Enlisted Female
draw om dotted     # Officers Male
draw em solid      # Enlisted Male
copy "army.cp" thru X
  next of at $1,$3
  next ef at $1,$5
  next om at $1,$2
  next em at $1,$4
X
copy thru % "$1 $2" size -3 at 60,$3 % until "XXX"
Enlisted Men 1200
Male Officers 140
Enlisted Women 12
Female officers 2.5
XXX
.G2
```



Eksempel på brug af eqn:

.EQ
erf (x) =
2 over sqrt pi int from 0 to x e sup { -t sup 2 } dt =
2 over sqrt pi sum from n=0 to inf { (-1) sup n x sup 2n+1 }
over { n! (2n + 1) }
.EN

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!(2n+1)}$$

Eksempel på brug af tbl:

```
.TS
box;
cf3 s s s.
Composition of Foods

-
.T&
c | c s s
c | c s s
c | c | c | c.
Food <TAB> Percent by Weight
\^ <TAB>
\^ <TAB> Protein <TAB> Fat <TAB> Carbo-
\^ <TAB> \^ <TAB> \^ <TAB> hydrate

-
.T&
l | n | n | n.
Apples <TAB> 0.4 <TAB> 0.5 <TAB> 13.0
Halibut <TAB> 18.4 <TAB> 5.2 <TAB> ...
Lima beans <TAB> 7.5 <TAB> 0.8 <TAB> 22.0
Milk <TAB> 3.3 <TAB> 4.0 <TAB> 5.0
Mushroom <TAB> 3.5 <TAB> 0.4 <TAB> 6.0
Rye bread <TAB> 9.0 <TAB> 0.6 <TAB> 52.7
.TE
```

Composition of Foods			
Food	Percent by Weight		
	Protein	Fat	Carbo- hydrate
Apples	0.4	0.5	13.0
Halibut	18.4	5.2	...
Lima beans	7.5	0.8	22.0
Milk	3.3	4.0	5.0
Mushroom	3.5	0.4	6.0
Rye bread	9.0	0.6	52.7

VOVV OV

Nummer 5

4. februar 1988

I dette nummer:

Indsættelse af stoppunkter i ORACLE-demostrationer
NIOC og SIOC2 med 64 kanaler til system V
Opfordring

1. Indsættelse af stoppunkter i ORACLE-demostrationer

Når man anvender »keystroke«-filer i forbindelse med SQL*Forms og runform vil en demonstration normalt forløbe for hurtigt. Derfor er det efterfølgende program »stopdemo« udviklet til at indsætter stoppunkter således at demonstrationen genoptages ved et vilkårligt tastetryk. Første trin i forberedelsen af en demonstration er indsættelse af stoppunkter med ESC eller Func. De følgende kald skal dog komme forud:

```
$ sqlforms -e keystrokes scott/tiger
```

```
$ runform -e keystrokes scott/tiger
```

Ved indsættelsen af stoppunkterne skal man ignorere fejlmeddelelsen og ikke kvitere med retur. Herefter kan demonstrationen genspilles på følgende måde (bruger h.h.v. ESC=0x1b og Func=0x00).

```
$ stopdemo 0x1b < keystrokes | sqlforms scott/tiger
```

```
$ stopdemo 0 < keystrokes | runform scott/tiger
```

OBS! Der er en række forhindringer som resulterer i et totalt stop af læsningen af »keystroke«-filen: SQL*Forms's makrokode CALL og CALLQRY, valget RUN fra SQL*Forms hovedmenu og formodentlig andre uopdagede steder. Læg også mærke til at ikke alle demonstrationer kan gentages, f.eks. sletning af en trigger vil ikke gå godt to gange i træk.

Programmet »æder« stoptegnet, der angives som parameter. Læg mærke til at stoptegnet er en oktet (byte), hvor funktionstaster optager to oktetter, f.eks f6 ~ 0x01 0x45.

```
/* stopdemo.c  
Creates breakes in keysstroke files by reading a char from the keyboard */
```

```
#include <stdio.h>
```

```
main(ac,av)  
int ac;  
char *av[];  
{  
FILE *tty;  
int c, s;  
  
if (ac != 2) {
```



```
printf("usage: stopdemo <stopchar> # dec, oct or hex");
exit(1);
}

sscanf(av[1], "%i", &s);
/* printf("Stop: %c0, s); */
tty = fopen("/dev/tty", "r");

for(;(c=getchar()) != -1;) {
    if (c == s) {fflush(stdout);fgetc(tty);continue;};
    putchar(c);
}
}
```

Programmets kildetekst »stopdemo.c« oversættes med:

```
$ cc -o stopdemo stopdemo.c
```

Ingen garenteres mod fejl, dog modtages ros.

hla [BP]

2. NIOC og SIOC2 med 64 kanaler til system V

Inden længe vil NIOC og SIOC2 være klar med 64 kanaler/256 vinduer (mod nu 32 kanaler/64 vinduer). Samtidig kommer der nye versioner af chhw og config.

SIOC2

Ved overgang til den nye version flyttes paralleporten fra kanal 31 til kanal 63. I den normale situation, hvor hver SIOC2 driver ét vægpanel, får man altså rådighed over 32 serielle linier plus den parallelle port. I den udvidede situation, hvor hver SIOC2 driver to vægpaneler, får man rådighed over 63 serielle linier plus den parallelle port. Det skal understreges, at der kun kan tvinges ca 10.000 tegn pr. sekund gennem SIOC2-en til fordeling mellem de tilsluttede enheder.

NIOC

Ved overgang til den nye version vil NIOC-ens netkonfiguration gå tabt! Systemadministratoren skal derfor være klar med sine konfigurationsfiler. Den nye version bliver fælles for 68000/68020 operativsystem med/uden streams på en NIOC med 0.5/1.0 Mbyte programlager monteret.

NIOC-en vil *automatisk* køre i en reduceret udgave med 32 kanaler/96 vinduer, hvis én af følgende betingelser er opfyldt:

1. NIOC-en har 0.5Mbyte programlager (ældre leverancer).
2. Kun de første 32 kanaler/96 vinduer er konfigureret med chhw (herved frigøres 0.5Mbyte programlager til fremtidige kommunikationsprotokoller).

I fremtiden kan man altså - *alene med chhw* - vælge mellem enten at udnytte 64 ttykanaler eller nøjes med 32 ttykanaler og derved få mere plads til kommunikationsprotokoller.

es [BP]

3. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes - brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse *ct*).

ct [BP]

VOVVOV

Nummer 6

8. marts 1988

I dette nummer:

Valg af MCU ved programstart
C++
Opfordring

1. Valg af MCU ved programstart

Vi ved alle at en af de vigtigste egenskaber ved Supermaxen er muligheden for at udnytte flere CPU'er. De fleste Supermaxer er udstyret med to eller flere MCU'er (Main Computing Units – dvs. de CPU'er som udfører de egentlige beregninger i modsætning til IOC'erne).

Når et program bliver startet – eller mere præcist: når `exec()` systemkaldet bliver udført – vælger styresystemet en MCU til det nye program. Algoritmen i dette MCU-valg er uhyre enkel: MCU'erne vælges cyklisk, således at det første program kører på den første MCU, det næste program kører på den næste MCU, og så fremdeles. Derudover har brugeren selv mulighed for at styre MCU-valget, enten ved hjælp af `mcu-mask` eller ved hjælp af `set_parm()` systemkaldet før `exec()`.

Det bliver jævnligt sagt at denne metode at vælge MCU'er på er for primitiv, og at man bør finde på en bedre metode. Lad mig blot nævne to problemer med den gældende metode:

1. Antag at man kører med 2 MCU'er, og al aktivitet består i at et lille program starter et stort program, hvorpå et lille program starter et stort program, hvorpå et lille program starter et stort program, osv. I dette tilfælde vil alle de store programmer komme på samme MCU, hvis ikke brugeren eller programmøren har pillet ved `mcumask`.

2. Hvis man konfigurerer en Supermax med én MCU med 2 MB lager og en anden MCU med 8 MB lager, har man ingen mulighed for at sikre sig at de fleste programmer havner på den MCU med mest lager.

Det skal ikke nægtes at den gængse metode til tider kan være uhensigtsmæssig, og den egentlige grund til at jeg skriver denne artikel er da også at jeg gerne vil opfordre folk til at komme med idéer til forbedringer. Men tillad mig først at nævne en lang række problemer som vil foregribe mange af de forslag der vil dukke op:

Hvis to brugere eksekverer samme program, skal disse to inkarnationer af programmet så ligge på samme MCU? Tja, hvis de ligger på samme MCU vil koden blive genbrugt, og det giver selvfølgelig bedre lagerudnyttelse. Men hvis de to programmer er meget CPU-belastende vil de genere hinanden frygteligt, så det er måske alligevel bedre at lægge dem på hver sin MCU. Og hvad hvis der er tredive brugere der eksekverer det samme program? Skal de så køre på samme MCU så man kan genbruge koden maksimalt?

Vi skal sikre en jævn belastning af MCU-ressourcerne, men hvad menes der egentlig med det? Når man skal starte et nyt program, skal det så køre på den MCU der har mest ledigt hovedlager? Eller på den MCU der har mest ledigt hovedlager + swaplager? Eller på den MCU hvis CPU inden for de sidste to minutter har været mindst aktiv? Eller på den MCU hvis FPU (flydende regnekreds) inden for de sidste to minutter har været mindst aktiv? Eller på den

MCU der inden for de sidste to minutter har swappet mindst? Eller på den MCU hvis hovedlager p.t. er mindst fragmenteret? Eller måske skal man udregne et vægtet gennemsnit, så man tager 30% hensyn til CPU-tidsforbrug, 10% hensyn til FPU-tidsforbrug, 40% hensyn til hovedlagerforbrug og 20% hensyn til swaplagerforbrug?

De forslag jeg i tidens løb har fået har alle haft én af to fejl. Enten forventes det at styresystemet skal kunne se ind i fremtiden (forslag af typen: »Hvis det program der skal til at køre har tænkt sig at bruge meget CPU-tid, skal man vælge på baggrund af CPU-tidsforbrug, men hvis programmet har tænkt sig at allokere meget lager, skal man vælge på baggrund af lagerforbrug«). Eller også er algoritmerne så komplicerede at en væsentlig del af styresystemets tid vil gå med at vælge en MCU.

Jeg skal ikke lægge skjul på at min egen holdning er den, at der ikke gives nogen generel løsning på problemet. Jeg tror at uanset hvilken løsning man vælger, vil den hjælpe i visse situationer og forværre andre situationer. Den metode vi bruger nu udmærker sig ved at være enkel og hurtig, og den virker glimrende i de fleste tilfælde. Når man så kobler dette sammen med mulighed for at brugeren eller programmøren selv kan styre MCU-valget, er man faktisk nået meget langt.

Den eneste forbedring jeg umiddelbart kan se nogen fremtid i, består i at man erstatter det cykliske valg af MCU'er med en tilfældighedsgenerator. Derved vil man få løst det problem som er nævnt under punkt 1 ovenfor. Men som sagt: Jeg modtager meget gerne forslag til forbedringer.

ct [BP]

2. C++

C++ er navnet på et såkaldt objekt-orienteret programmeringssprog. C++ er vores gode gamle C med en række nye faciliteter, som skulle gøre det muligt at skrive bedre/pænere/rigtigere programmer.

2.1 Hvad kan C++?

Alt hvad I kender fra C findes også i C++. Derudover findes en række nye faciliteter

som f.eks.:

- Mulighed for at definere operatorer (+, -, *, etc.) på brugerdefinerede typer.
- Mulighed for »data hiding«, hvilket vil sige at man kan tvinge programmører til kun at accesse datatyper på en veldefineret og civiliseret måde.
- Funktions-overloading, afgjort på kompileringstidspunktet.
- Funktions-overloading, afgjort på eksekveringstidspunktet.
- Inline-funktioner.
- Konstanter.
- Referencevariable.
- Og meget andet.

2.2 Hvordan kører man C++?

C++-oversætteren oversætter til almindelig C-kode, som derpå hældes gennem den almindelige C-compiler.

2.3 Hvor kan jeg læse mere?

C++ er beskrevet i »The C++ Programming Language« af Bjarne Stroustrup, Addison-Wesley Publishing Company, 1986.

2.4 Hvordan får jeg fat i C++?

Vi har i basisprogrammefordelingen en C++-oversætter. Vi er i færd med at undersøge om vi inden for rammerne af vores licens har lov til at distribuere den inden for DDE. Hvis vi ikke må det, vil vi hurtigst muligt bringe licensbetingelserne i orden. C++-oversætteren vil, når disse problemer er løst, kunne fås fra Helle Feldthaus.

2.5 Seminar

Undertegnede er villig til at holde et (særdeles uformelt) internt seminar på et par timer om C++. Betingelsen er at en eller anden gider ordne det praktiske (fastlæggelse af tidspunkt, reservation af mødelokale, indkaldelse af interesserede m.m.).

ct [BP]

3. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle (*osse jer uden for BP!*) særdeles

velkomne til at sende mig indlæg til bladet.
Indlæg på magnetisk form foretrækkes –
brug fx en diskette, eller send mig artiklen
med UUCP-mail (postadresse ct).

ct [BP]

VOVVOV

Nummer 7

22. april 1988

I dette nummer:

Nyt om UUCP-adresser
Dimensionering af UPS
En lille opgave
Opfordring

→ KAN

Burde udsendes
Som note til sælgende
medarbejdere og ikke
kun til VOV-VOV

1. Nyt om UUCP-adresser

Der er for tiden ved at ske nogle ændringer på UUCP-nettet, som har visse konsekvenser for vores UUCP-adresser.

For det første: Når vi får elektronisk post ude fra den store verden er denne post hidtil blevet leveret via en maskine ved navn »diku«. Dette er lavet om så vi nu får post fra en maskine ved navn »dkuug«.

For det andet: Hele det verdensomspændende UUCP-net er efterhånden lagt om så alle forstår @-adresser. Det er med andre ord ikke længere nødvendigt at bruge !-adresser.

For det tredje: @-adresser i Danmark bør fremover ende på ».dk« i stedet for ».uucp«.

Hvad betyder så alt dette?

Hidtil har det været god latin at opgive sin postadresse på formen »ct@dde.uucp« eller »...!mcvax!diku!ddelct« (undskyld min ubeskedne brug af mig selv som eksempel). Disse to adresseformer er stadigvæk i orden, også selv om posten kommer via »dkuug« i stedet for »diku«. Men der er ingen garanti for at de gamle adresseringsformer vil blive forstået i al fremtid. I bør derfor fremover fortrinsvis bruge adresser af formen »ct@dde.dk«, altså uden udråbstegn og med ».dk« i stedet for ».uucp«.

ct [BP]

2. Dimensionering af UPS

DDE har i knap et år haft to UPS (Uninterruptable Power Supply) eller No-break anlæg på programmet, 9132 No-break system 1000 VA og 9136 No-break system 2000 VA. De to anlæg er udviklet og produceret af SILCON A/S i Kolding, og de er begge opbygget i et Slimline kabinet. Følgende »tommelfingerregel« vedrørende dimensionering af No-break anlæg kan (og bør) anvendes ved salg og installation af No-break anlæg.

Maskintype	Antal	
	1000 VA	2000 VA
1 Compact	1	
1 Slimline	1	
1 Vertical		1
1 Slimline + 1 Diskkabinet		1
1 Vertical + 1 Diskkabinet		1
2 Vertical		2
2 Vertical + 1 Diskkabinet		2
2 Vertical + 2 Diskkabinet		2
Rackmodel:		
1 Kortkasse + 1 Diskkasse		1
1 Kortkasse + 2 Diskkasse		1
1 Kortkasse + 3 Diskkasse	1	1
1 Kortkasse + 4 Diskkasse		2
2 Kortkasse + 1 Diskkasse		2
2 Kortkasse + 2 Diskkasse		2
2 Kortkasse + 3 Diskkasse		2

Hvis man følger ovennævnte regler ved dimensionering af No-break anlæg, er man sikker på at No-break anlægget er

tilstrækkelig stort, selv ved fuld udbygning af maskinen.

Opmærksomheden henledes på installationskravene, som er beskrevet i datablad **mu01 No-break system 1000VA/2000VA**. Disse krav (separat netgruppe og større sikringer) vil normalt betyde en ekstra udgift for kunden, idet der kræves elektrikerbesøg.

I øvrigt henvises til en udmærket beskrivelse af operativsystemets behandling af powerfail interrupt i Vovov nr. 4.

lbp [MU]

3. En lille opgave

Nedenstående program vandt præmie som »mest ulæselige én-linies C-program« i en international konkurrence i 1987. Prøv først at gætte hvad det skriver, derpå kan I jo prøve at taste det ind og se om I havde ret.

```
main() { printf(&unix["\21%six\12\0"],(unix){"have"}+"fun"-0x60); }
```

ct [BP]

4. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle (*osse jer uden for BP!*) særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes — brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse *ct*).

ct [BP]

VOVVOV

Nummer 8

11. juli 1988

I dette nummer:

Spoolmodeller til intern brug
ISO 8859/3 og 8859/4
Opfordring

1. Spoolmodeller til intern brug

Da de — mig bekendt — eksisterende spoolmodeller ikke kontrollerer om sidst udskrevne fil afsluttedes med formfeed, men ukritisk afslutter udskriften med formfeed, har jeg ændret model/dumb til at checke for dette.

Modellerne forekommer i to udgaver, model/iq200 og model/iq200-garbag. Førstnævnte udskriver for/bag-side og sidstnævnte gør det ikke.

Der checkes både for tegn 12 og haksekvenserne X>, E> og E?;?>.

Modellerne er installeret på AS maskinerne Giraffen og Missen, og kan kopieres fra disse. Ellers er nedennævnte villig til formidling.

shc [AS]

2. ISO 8859/3 og 8859/4

I april i år udkom to ny skud på ISO-8859-stammen. ISO 8859 er, som mange vil vide, en fællesbetegnelse for en række udvidelser af ASCII-tegnsettet. På Supermax anvender vi normalt ISO 8859/1, men af hensyn til eventuelle interesserede findes de to nye standarders tegnsættabeller vedlagt dette nummer af Vovvov.

ISO 8859/3 er rettet mod følgende sprog: Afrikaans, catalansk, engelsk, esperanto, hollandsk, italiensk, maltesisk, spansk, tyrkisk og tysk. Der er altså *meget groft* tale om landende i Sydeuropa. For DDE er det

vel specielt interessant at tyrkisk er med i denne standard.

ISO 8859/4 er rettet mod følgende sprog: Dansk, engelsk, estlandsk, finsk, grønlandsk, laplandsk, lettisk, litauisk, norsk, svensk og tysk. Her er der altså tale om de nordeuropæiske sprog.

Af øvrige 8859-standarder kan nævnes:

- 8859/1: Vesteuropæisk
- 8859/2: Østeuropæisk
- 8859/5: Kyrillisk (dvs. russisk m.m.)
- 8859/6: Arabisk
- 8859/7: Græsk
- 8859/8: Hebraisk

ct [BP]

3. Opfordring

Vovvov er et internt teknisk tidsskrift for DDE. I er alle (*osse jer uden for BP!*) særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes — brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse ct).

ct [BP]

Table 2 — Code table of Latin alphabet No. 3

				b ₁	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
				b ₂	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
				b ₃	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b ₄	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
					00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
b ₁	b ₂	b ₃	b ₄																	
0	0	0	0	00			SP	0	@	P	`	p			NBSP	°	À	à		
0	0	0	1	01			!	1	A	Q	a	q			Ħ	ħ	Á	Ñ	á	ñ
0	0	1	0	02			"	2	B	R	b	r			˘	²	Â	Ò	â	ò
0	0	1	1	03			#	3	C	S	c	s			£	³		Ó		ó
0	1	0	0	04			\$	4	D	T	d	t			¤	'	Ä	Ô	ä	ô
0	1	0	1	05			%	5	E	U	e	u			μ	˘	Ĉ	Ĝ	ĉ	ĝ
0	1	1	0	06			&	6	F	V	f	v			Ĥ	ĥ	Ĉ	Ö	ĉ	ö
0	1	1	1	07			'	7	G	W	g	w			Š	-	Ç	×	ç	÷
1	0	0	0	08			(8	H	X	h	x			"	,	È	Ĝ	è	ĝ
1	0	0	1	09)	9	I	Y	i	y			İ	ı	É	Ù	é	ù
1	0	1	0	10			*	:	J	Z	j	z			Ş	ş	Ê	Ú	ê	ú
1	0	1	1	11			+	;	K	Ł	k	ł			Ǧ	ǧ	Ë	Û	ë	û
1	1	0	0	12			,	<	L	ł	l	ł			Ĵ	ĵ	Ì	Û	ì	ü
1	1	0	1	13			-	=	M	Ź	m	ź			SHY	½	Í	Ů	í	ů
1	1	1	0	14			.	>	N	ˆ	n	˜					Î	Ŝ	î	ŝ
1	1	1	1	15			/	?	0	_	o	˘			Ž	ž	İ	ß	ï	˙

Table 2 — Code table of Latin alphabet No. 4

					b.	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
					b.	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1	
					b.	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
					b.	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
					00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15		
b.	b.	b.	b.																			
0	0	0	0	00			SP	0	á	À	·	p			NBSP	°	Ā	Ð	ā	đ		
0	0	0	1	01			!	1	À	Q	a	q			Ą	ą	Á	Ń	á	ñ		
0	0	1	0	02			"	2	B	R	b	r			κ	.	Â	Ō	â	ô		
0	0	1	1	03			#	3	C	S	c	s			Ŕ	ŕ	Ã	Ķ	ã	ķ		
0	1	0	0	04			\$	4	D	T	d	t			α	'	Ä	Ô	ä	ô		
0	1	0	1	05			%	5	E	U	e	u			İ	ı	Æ	Ö	æ	ö		
0	1	1	0	06			&	6	F	V	f	v			Ł	ł	Æ	Ö	æ	ö		
0	1	1	1	07			'	7	G	W	g	w			Š	š	ı	×	ı	÷		
1	0	0	0	08			(8	H	X	h	x			"	,	Č	Ø	č	ø		
1	0	0	1	09)	9	I	Y	i	y			Š	š	É	Û	é	Û		
1	0	1	0	10			*	:	J	Z	j	z			Ě	ě	Ę	Ú	ę	ú		
1	0	1	1	11			+	;	K	Ł	k	ł			Ğ	ğ	Ë	Û	ë	û		
1	1	0	0	12			,	<	L	\	l				ƒ	ƒ	È	Ü	è	ü		
1	1	0	1	13			-	=	M]	m	}			SHY	Ń	Í	Ū	í	ū		
1	1	1	0	14			.	>	N	^	n	~			Ž	ž	Î	Ū	î	ū		
1	1	1	1	15			/	?	0	-	o				̄	Ń	İ	ß	ı	·		

VOVVOV

Nummer 9

4. august 1988

I dette nummer:

Konfigurering af Supermax
stty2
UUCP-nettet
Nye 5¼" floppy formater
Nyt distributionsformat på tapes
Opfordring

1. Konfigurering af Supermax

Da der har vist sig et vist behov for dels at udregne hvordan en Supermax skal konfigureres rent hardwaremæssigt, dels udregne størrelsen af de enkelte operativsystem parametre der kan justeres med chhw programmet, har undertegnede forsøgt at opstille nogle generelle tommelfingerregler for udregning af ovennævnte.

Da disse regler er beskrevet i en manual der i øjeblikket er til oversættelse (fra dansk til engelsk) vil jeg være meget glad for eventuelle kommentarer til disse regler, inden de slipper officielt ud. Jeg har derfor plukket reglerne ud fra manualen og oversat den tilhørende tekst til engelsk (af hensyn til IS).

Dokumentet findes bagest i dette nummer af Vovvov.

PS: Jeg er *fuldt* ud klar over at reglerne ikke dækker alle mulige specialsituationer, men dette har heller ikke været intentionen.

ken [BP]

2. stty2

Programmet *stty2* understøtter en række terminalopsætninger som ikke er præcist specificeret i dokumentationen. Dette rådes der bod på nedenfor:

Man kan specificere opsætningerne *sane*, *sane8*, *dde* og *dde_ctl*. Disse opsætninger bevirker sætning af følgende flag (blanke

indgange i tabellen betyder at de pågældende flag ikke berøres):

sane	sane8	dde	dde_ctl
		line 1	
		viin	
		vtout	
intr DEL	intr DEL	intr '^c'	intr '^c'
quit '' '	quit '' '	quit '' '	quit '' '
erase #	erase #	erase '^h'	erase '^h'
kill @	kill @	kill undef	kill undef
eof ''d'	eof ''d'	eof ''d'	eof ''d'
eol NUL	eol NUL	eol NUL	eol NUL
swtch NUL	swtch NUL	swtch NUL	swtch NUL
		att '^b'	att '^b'
parenb	-parenb	parenb	
-parodd	-parodd	-parodd	
cs7	cs8		
cread	cread	cread	
-clocal	-clocal	-clocal	
-ignbrk	-ignbrk	-ignbrk	
brkint	brkint	brkint	
ignpar	-ignpar	ignpar	
		-parmrk	
-inpck	-inpck	-inpck	
istrip	-istrip	istrip	
-inlcr	-inlcr	-inlcr	
-igncr	-igncr	-igncr	
icrnl	icrnl	icrnl	
-iuclic	-iuclic	-iuclic	
ixon	ixon	ixon	
-ixoff	-ixoff	-ixoff	
isig	isig	isig	

sane	sane8	dde	dde_ctl
icanon	icanon	icanon	
-xcase	-xcase	-xcase	
echo	echo	echo	
-echoe	-echoe	-echoe	
echok	echok	echok	
-echonl	-echonl	-echonl	
-nofish	-nofish	-nofish	
opost	opost	opost	
-olcuc	-olcuc	-olcuc	
onlcr	onlcr	onlcr	
-ocml	-ocml	-ocml	
-onocr	-onocr	-onocr	
-onlret	-onlret	-onlret	
-ofill	-ofill	-ofill	
-ofdel	-ofdel	-ofdel	
nl0	nl0	nl0	
cr0	cr0	cr0	
tab0	tab0	tab3	
bs0	bs0	bs0	
vt0	vt0	vt0	
ff0	ff0	ff0	

ct [BP]

3. UUCP-nettet

I løbet af de sidste par uger er der sket ting med UUCP-nettet.

For det første kører netnews nu atter efter lang tids pause. En fejl i NIOC'en er blevet rettet, og det betyder at det atter er forsvarligt at belaste maskinerne med de mange nyhedsartikler.

For det andet blev der i forgårs (den 2. august) fundet et grufuld fejl på dde-maskinen. Et forsøg på at lave en smart retelse i mail-scriptet har været skyld i at vi i et par måneder har mistet store mængder udefra kommende post. Meget post, der er kommet ude fra den store verden til DDE inden for de sidste par måneder, er simpelthen blevet hældt direkte ned i et stort sort hul!

Denne fejl burde være rettet nu.

ct [BP]

4. Nye 5½" floppy formater

I forbindelse med operativsystem 520.4C og 500.8C er der indført læsning/skrivning af nye floppy formater. Med disse nye formater er vi i stand til at udveksle data på 5½" floppy disketter med PC (XT), de fleste UNIX maskiner (X/OPEN) samt IBM system 34 (ABI). I forbindelse med 40 tracks disketter kan der opstå det problem, at vi efter at have skrevet en sådan diskette på supermax (80 tracks drev), ikke er i stand til at læse den på en PC (XT 40 tracks drev). Som regel går det godt, men vi kan ikke garantere det. Det vil altid være muligt at læse en sådant diskette på en PC (AT 80 tracks drev).

Vores standard format er stadig 560K (DDE-standard), og vi understøtter ikke formateringen af andre formater.

For at kunne skrive/læse MS-DOS disketter på filformat findes en samling programmer kaldet MTOOLS. Disse programmer vil blive frigivet fra BP i uge 32.

De fysiske kanaler på DIOC2 har nu følgende funktioner:

Channel	Function
0	Reserved
1	First 1 Megabyte 8" floppy drive
2	Second 1 Megabyte 8" floppy drive
3	First 560 Kbyte 5¼" floppy (70 tracks 16 sectors)
4	Second 560 Kbyte 5¼" floppy (70 tracks 16 sectors)
5	First IBM compatible 8" floppy drive
6	Second IBM compatible 8" floppy drive
7	Streamer tape
8	First hard disk on first controller
9	Second hard disk on first controller
10	First hard disk on second controller
11	Second hard disk on second controller
12	First hard disk on third controller
13	First hard disk on fourth controller
14	First hard disk on fifth controller
15	First hard disk on sixth controller
16	Mag tape (Low density) / Optical disk
17	Mag tape (High density) / Optical disk
18	Not used
19	Not used
20	First 360 Kbyte 5¼" floppy (40 tracks 9 sectors)
21	Second 360 Kbyte 5¼" floppy (40 tracks 9 sectors)
22	First 720 Kbyte 5¼" floppy (80 tracks 9 sectors)
23	Second 720 Kbyte 5¼" floppy (80 tracks 9 sectors)
24	First 320 Kbyte 5¼" floppy (40 tracks 8 sectors)
25	Second 320 Kbyte 5¼" floppy (40 tracks 8 sectors)
26	First 640 Kbyte 5¼" floppy (80 tracks 8 sectors)
27	Second 640 Kbyte 5¼" floppy (80 tracks 8 sectors)

Mbyte, -3 for 120 Mbyte.

jup [MUDV]

6. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle (**osse jer uden for BP!**) særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes – brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse ct).

ct [BP]

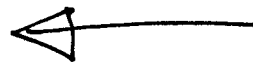
jup [MUDV]

5. Nyt distributionsformat på tapes

Fra 1-9-88 vil alle tapes genereret i Klokkerholm skifte format fra QIC-11 til QIC-24. Det betyder, at alle kunder der har 20 Mbyte streamere ikke længere vil kunne læse standardtapes. Hvis man ønsker tapes i det gamle format skal det angives ved bestillingen. Denne ændring er nødvendig i forbindelse med introduktion af en ny 120 Mbyte streamer, som ikke understøtter det gamle format. Oversigt over formater på ½" tapes:

20 Mbyte streamer: læs/skriv QIC-11
45 Mbyte streamer: læs/skriv QIC-24
120 Mbyte streamer: læs/skriv QIC-120, læs QIC-24

Det er vigtigt at definere streameren rigtigt med *chhw*, dvs. -1 for 20 Mbyte, -2 for 45



← Bemærk venligst dette ved fremtidige bestillinger

SUPERMAX GENERAL CONFIGURATION RULES

Supermax System V

ken 07.06.88

This paper describes a number of general rules made to make it more easy to configurate the Supermax computer with the 'correct' number of hardware modules and calculate the values of the operating system parameters.

The rules are split into two main sections, the first section containing rules related to calculating the number of MCU's, DIOC's, SIOC's, etc..., the second section containing rules related to calculation of the operating system parameter values set by the *chhw* program.

Please note that all the rules in this paper are only rules of thumb and therefore are not totally exact, but experience shows that the rules normally give a very good indication of the number of modules and values of the operating system parameters.

The rules are valid for all the current versions of the System V operating system for the Supermax.

HARDWARE CONFIGURATION RULES:

The following rules are made in an attempt to make it more easy to determine the number modules a Supermax should be configurated with. The main parameters used by the rules are the number of users on the machine and the total sum of the sizes of the harddisks in MB.

The hardware configuration rules are general and should normally be used by sales people to make an estimate of the number of each of the various modules a Supermax should be equipped with.

Module	Calculation rule
CPU 0101	MCU = USER / 8
CPU 3400	MCU = USER / 20
RAM 3x00	MEM = (USER / MCU * 0.4) + 1.5
DIOC 1100	DIOC = ((MCU / 2) + (SUM / 250MB)) / 2
SIOC 0300	SIOC = (TERM + PRINT) / 8
SIOC 3600	SIOC-2 = (TERM + PRINT) / 32
NIOC 1600	NIOC = (TERM + PRINT) / 32
CIOC 0900	CIOC = (Physical channels) / 2

The following abbreviations are used:

USER: The total number of users.
SUM: The total size of all the hard disks (MB).
TERM: The total number of terminals.
PRINT: The total number of printers.
MCU: The total number of MCU'es.

OPERATING SYSTEM CONFIGURATION RULES:

The Operating system configuration rules are made to determine the value/size of the various parametres used by the operating system kernel and set by the *chhw* program.

The rules are general and should normally be used by system administrators, technicians and other people responsible for installing and adjusting the operating system on the Supermax, to calculate the values of the various parametres.

The rules can be used to check one or more parameters of an existing configuration or can be used to calculate all the values in which case the rules should be used in sequence from the top to the bottom of the schematics.

Note that for some of the parameters the rule is just returning a constant value. Those parameters should normally be set to the constant value but in 'special' cases this value should be increased to satisfy requirements from 'special' applications. For example, when running various Cobol programs where a lot of record locking is used, the number of lock elements (LOCKELEMENTS) should be increased.

GLOBAL PARAMETRES:

Resource	Calculation rule
LOCKELEMENTS	10
MESSAGES	10
MAX I/O	32
GLOBALS	$(USER * 3) + 15$
FILES	$GLOBALS * 2$
OPENS	$FILES * 3$
SHARED	$> = 10$
SEMAPHORE	10

LOCAL PARAMETRES:

Resource	Calculation rule
ALLOWED MEM	1 MB
LOCALS	$GLOBALS / MCU + 10$
PART DESC	$3 * LOCALS$
TEXT DESC	$1 * LOCALS$
ITEM	$(FILES/MCU * 4 \text{ KBytes}) + (LOCALS * 1 \text{ KBytes})$

The abbreviations used are the same as for the hardware configuration rules.

VOVVOV

Nummer 10

8. november 1988

I dette nummer:

Ny *cpio* og *bcpio*
Hurtig C programmering
Power Tools
UNIX® System V Release 4.0
μEMACS – én editor til alle opgaver
Opfordring

1. Ny *cpio* og *bcpio*

Med basic utilites rel. 2.5 udkommer en *cpio*, der *ikke* er kompatibel med den tidligere version.

Problemet er at AT&T fra Rel.3.1 har indført en option »M«, der angiver en meddelelse ved båndskift. »M« har imidlertid hidtil været en speciel DDE option til buffering ved kopiering til/fra streamer tapes. For at undgå misforståelser bliver det sådan at den nye *cpio* hverken supporterer AT&T's eller DDE's option »M«.

Dette betyder at *cpio* ikke vil kunne anvendes ved:

- kopiering over flere medier,
- kopiering til/fra streamer tapes.

I disse situationer skal man benytte en ny utility *bcpio*. *bcpio* er et script der »piper« *cpio* gennem *streamdrv*.

DVS. Alle »backup scripts«, der benytter *cpio* med option »M«, og alle, der kopierer over flere medier, skal ændres til at bruge *bcpio*.

coh [BP]

2. Hurtig C programmering

Da vi i BP (specielt operativsystemgruppen) i flere år haft benchmark-pisken over os, har vi brugt en del tid på at finde ud af hvordan man får C programmer til at køre hurtigt. For at andre kan få glæde af denne viden,

har jeg lavet et skrift »Programmering af Hurtig C kode«, der beskriver nogle af de teknikker vi har anvendt. Skriftet handler *ikke* om algoritme-valg, da der er skrevet utallige bøger om dette emne, men om hvordan man får den valgte algoritme til at køre hurtigt. Skriftet henvender sig til programmører der i forvejen har kendskab til C, og kan rekvireres hos undertegnede (Thomas).

tpo [BP]

3. Power Tools

This is an introduction for a new generation of in-house power tools.

This distribution includes high end power tools exploiting the newest concurrent performance blasting technology. Products not yet released are:

cmake Concurrent *make* with output synchronization, dynamic MCU load control and many unknown features. For heavy jobs with many independent tasks to be executed in parallel, the real time will be reduced according to the numbers of MCU's, resulting in a real time less than user time. Normally placed in /sbin as *make*. Having /sbin first in \$PATH will result in a major productivity increase. (Available, separate manual page included.)

- ccc Concurrent cc with a pipe from precompiler to compiler, a smart tool for a smart senior programming consultant engineer. Normally placed in /sbin as cc. Having /sbin first in \$PATH will result in a major productivity increase. (Contact Helle Feldthaus.)
- index A fast symbol recognizer used for searching for symbol in large text bundles. This may be program texts, mail files, news files, telex, plain text — you name it. Identified symbols and related files names are stored in an Oracle database in normalized and indexed tables. (Prerelease exists.)

Orders may be placed immediately. Cash on delivery, bribery accepted.

Yours faithfully Henrik & Henrik (BP)
hal & hb [BP]

4. UNIX[®] System V Release 4.0

I slutningen af 1989 sender AT&T release 4.0 af UNIX System V ud på markedet. Denne release vil indeholde en hel del nye faciliteter, og til generel orientering vil jeg nedenfor redegøre for de væsentligste.

Release 4.0 vil være en sammenfletning af AT&T's System V, Microsofts XENIX og Berkeleys BSD 4.3. Dette betyder at mængden af kommandoer og funktioner er vokset betydeligt, dels fordi BSD har tilført mange nye egenskaber, dels fordi man ofte har været nødt til at tillade at en og samme ting bliver gjort på to forskellige måder.

Release 4.0 vil endvidere være i fuld overensstemmelse med POSIX-standarden P1003.1 fra IEEE og C-standarden X3J11 fra ANSI.

AT&T satser hårdt på release 4.0. De forsøger at få the Open Software Foundation (OSF) overtalt til at gøre denne release til basis for OSF's styresystem. Endvidere vil release 4.0 danne grundlaget for de såkaldte ABI'er (Applications Binary Interface) — mere herom senere.

Her følger lidt løst og fast om de nye faciliteter:

- Tre shells: Den gammelkendte Bourne shell, C shell, og den skærmorienterede Korn shell. Alle tre shells forsynes med »job control«, som skulle overflødiggøre det hidtidige vinduesprincip, der bl.a. kendes fra *sh/* og DDE's vinduesprogrammel.
- Ubegrænset antal åbne filer pr. proces.
- Mulighed for at mappe en del af en fil ind i hovedlageret og således arbejde direkte på filens data.
- Flere nye filsystemer:
 - Berkeleys Fast File System med filnavne på 255 tegn i stedet for 14.
 - NFS.
 - Et pseudofilsystem som i virkeligheden udpeger de enkelte processers lager.
- Symbolske links der kan definere synonymmer for filer.
- Streams-interface til terminaler, piper m.m.
- Én *getty* pr. maskine i stedet for én pr. terminal.
- TCP/IP.
- Remote Procedure Call. Gør det muligt at kalde subrutiner på andre maskiner.
- Ny signal-håndtering (igen).
- Visse real-time faciliteter.
- Bedre fejlmeddelelser fra standard-utilities.
- Nyt loadmodul-format.
- Linking af programmer på eksekveringstidspunktet i stedet for på oversættelsestidspunktet. Dette betyder at man kan distribuere en ny udgave af et bibliotek uden at behøve at gen-linke alle programmer.
- X Window System og NeWS.
- Nemmere systemadministration.

4.1 Brugerinterface

AT&T har fastlagt to standarder for hvorledes et applikationsprogram skal præsentere sig for brugeren.

Den ene standard gælder konventionelle terminaler. Vi i DDE har allerede et ganske ensartet brugerinterface som benyttes i fx. Supermax Tekst, kalendersystemet, Supermax Post osv. Nu har AT&T altså gået os i bedene og definerer deres egen standard for en sådan interface. Interesserede kan få en kopi af standarden hos mig.

Den anden standard er betydelig mere omfattende. Den gælder interface til grafiske skærme der kører fx X Window System. Denne interface kaldes OPEN LOOK. Det er A&T's (utivlsomt korrekte) formodning at mange fremtidige applikationsprogrammer vil blive skrevet til den slags grafiske skærme, og derfor er det vigtigt at få en standard for hvorledes disse programmer skal præsentere sig for brugeren. AT&T leverer dels en specifikation af denne interface, dels en række rutiner som skal lette kodningen af den slags programmer oven på X Window System eller NeWS. Dokumentation fås hos undertegnede.

4.2 ABI

I forbindelse med release 4.0 udsendes en række binære interface standarder, Applications Binary Interface (ABI). Meningen med disse er skabe et miljø hvor eksekverbare programmer kan flyttes fra én maskine til en anden, som vi kender det fra PC-verdenen. Naturligvis kan et program ikke flyttes fra en Intel CPU til en Motorola CPU, men inden for en given chip-type er flytbarhed mulig. I første omgang kommer der ABI'er til følgende chips: Intel 80386, Motorola 68020 & 68030, Motorola 88000, SPARC, MIPS, Clipper.

4.3 SVID

En ny udgave af SVID med de nye faciliteter forventes klar i første kvartal af 1990.

ct [BP]

5. μ EMACS – én editor til alle opgaver

Igennem længere tid har jeg (og nogle få andre) benyttet μ EMACS i stedet for den traditions bundne vi-editor under UNIX system V. μ EMACS er en public domain editor og i modsætning til mange andre public domain programmer er μ EMACS et

fortræffeligt produkt.

Hvis jeg kort skal opremse nogle få af μ EMACS's muligheder, så vil listen have følgende udseende:

- Man har forskellige typer af on-line hjælp bl.a. funktionstastoversigt, funktionsoversigt og en tutorial.
- Man er altid i »insert-mode«, dvs. man benytter specialtaster til aktivering af editingsfunktioner.
- Man har mulighed for at tilpasse μ EMACS's funktioner ved at arbejde i forskellige modes. Modes kan også kombineres. Der er flg. modes i μ EMACS:

WRAP ombryder indtastet tekst ved en højre margin.

VIEW ikke muligt at rette teksten.

CMODE sørger for korrekt indrykning og parentesmatch.

OVER nye indtastninger overskriver gammel tekst.

CRYPT filen (de-)krypteres (ved indlæsning/)når filen skrives ud på disk.

ASAVE filen gemmes automatisk efter et forudbestemt antal indsatte tegn.

EXACT eksakt tegnmatch ved søgestreng.

MAGIC benyt regulære udtryk i søgninger.

- Man kan arbejde på mange filer på én gang.
- Man kan dele skærmen op i »vinduer«, hvor man så kan vælge at arbejde på forskellige udsnit af den samme fil eller på helt andre filer.
- Hvert vindue har en status-linie, der fortæller hvilken mode man arbejder i, bufferens (og evt. filens) navn. Desuden kan man se om filen er blevet ændret siden den sidst blev gemt.
- Man kan naturligvis kopiere/flytte tekst fra et »vindue« til et eller flere andre. μ EMACS husker selvfølgelig hvor du/cursoren var ved skift af

vindue/buffer.

- Man kan til enhver tid få en liste over μ EMACS's funktioner og de taster de er bundet til (ved at udføre funktionen »describe-bindings«).
- Man kan til enhver tid få oplysning om hvilken funktion (hvis nogen), der er bundet til en bestemt taste (-sekvens).
- μ EMACS's funktioner har jeg forsøgt at binde så tæt opad DDE-standardblad nr. 5 (Funktionstaster) som muligt. Det er dog altid muligt (i løbet af 5-10 sekunder) at flytte en funktion fra en taste(-sekvens) til en anden.
- Man kan udføre samtlige μ EMACS-kommandoer uden brug af special/funktionstaster blot ved at angive funktionens navn.
- Man har mulighed for en liste over kommandoer/funktioner vedrørende et bestemt emne ved hjælp af kommandoen »apropos«.
- Man kan »optage« en tastesekvens som kan udføres vilkårligt mange gange (key-board macro).

Listen indeholder kun de funktioner som man må stille som minimum krav til et professionelt værktøj. μ EMACS indeholder også nogle spændende muligheder, der virkelig kan forøge produktiviteten for den enkelte medarbejder. Her tænkes specielt på et lille programmeringssprog indbygget i μ EMACS, der gør, at brugeren kan skrive nye makroer og nye kommandoer ved hjælp af gamle makroer/kommandoer, som kan benyttes på lige fod med μ EMACS's egne kommandoer. Dette bevirker, at μ EMACS kan dække den enkelte medarbejders behov i alle situationer.

Unix's *vi* har ét kæmpefortrin fremfor alle andre editorer, nemlig at den findes i alle Unix systemer. Filosofien er så, at man kan nøjes med at lære én editor at kende (man kan sikkert også vænne sig til sand i øjnene og eksamen). Dette monopol slår μ EMACS nu hul på for DDE-medarbejdere og kunder, idet det er lovligt at kopiere og distribuere μ EMACS til alle ikke-kommercielle formål.

Et andet fortrin *vi* har haft er, at den kan benytte såkaldte »tags«. Tags er en fil, hvor alle C-funktionsnavne og makroer for et

projekt findes sammen med et søgemønster og navnet på den fil, hvor funktionen er defineret. *Vi* kan ud fra tags-filen således finde frem til det sted, hvor funktionen er defineret. Det kan man naturligvis også i μ EMACS (man kan desuden re-tagge i tilfælde hvor funktionen/makroen er defineret flere gange!).

Jeg mener, at alle medarbejdere altid bør have en diskette ved hånden (også ude ved kunderne) med deres yndlingsværktøjer, således at de til enhver tid kan yde den hurtigste og bedste service. Værktøjsdisketten bør være en lige så naturlig ting som kørekort, sygesikringsbevis og dankort. μ EMACS vil absolut være at finde på din diskette!

Hvis du har fået lyst til at prøve μ EMACS og den ikke allerede er at finde på din maskine, så bed din systemadministrator om at kontakte mig. Manual og en kort hjælpeoversigt får du af din systemadministrator.

mhs [AS/A3]

6. Opfordring

Vovvov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes — brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse ct).

ct [BP]

NAME

cmake - maintain, update, and regenerate groups of programs, concurrently

SYNOPSIS

cmake [-f *makefile*] [-p] [-i] [-k] [-s] [-r] [-n] [-b] [-e] [-t] [-d] [-q]

DESCRIPTION

The *cmake* command is a concurrent version of the well known UNIX command *make*. *Cmake* executes commands in parallel when they are not forced to some specific execution order by the makefile or build in rules.

By exploiting the potential of parallelism *cmake* may run several times faster than *make* on a multi MCU Supermax. On a 3 MCU Supermax the time to make a program from several source files is typically reduced by a factor of 2 - 3.

A maximum of 3 commands per MCU are run concurrently and new commands are started on the MCU where fewest commands are already running (in order to balance the load among the MCU's).

Output (standard output and standard error) from a command sequence is postponed until the command sequence has completed execution. Consequently output from different concurrent commands is not intermixed, but appears as if the commands were executed sequentially.

FILES

/bin/*cmake* and /sbin/*make*

SEE ALSO

make(1)

CAVEATS

Although great care has been taken to maintain compatibility with *make* it happens that *cmake* produces unexpected results due to the concurrent execution of commands. Expect problems, if:

Some particular order of execution is assumed although this order is not deduceable from the makefile and the build in rules.

Two or more commands write to the same file concurrently (eg. *lex* and *yacc* always use the same filename for output).

VOVVOV

Nummer 11

18. januar 1989

I dette nummer:

ANSI-C / DIAB cc
X/OPEN Security Guide
ISO-DIS 8859/9
Opfordring

1. ANSI-C / DIAB cc

Igennem nogle år har det amerikanske standardiseringsråd, ANSI, arbejdet med at standardisere C. På internationalt plan har den internationale standardiseringsorganisation, ISO, taget ANSI's arbejde op, og målet er snarest at få en international C-standard.

ANSI-C har nogle nyttige tilføjelser til gængs C, og visse af disse tilføjelser er allerede indarbejdet i den *dcc*-compiler fra DIAB som nu bruges mange steder internt i DDE. ANSI-C-standarden kan lånes hos mig, men nedenfor vil jeg redegøre for en del af de nyskabelser som allerede findes i *dcc*.

1.1 Prototyping

En gængs C-rutineerklæring ser fx. således ud:

```
alfa(a,b,c,d)
    int a;
    double b, c;
    char *d;
```

I ANSI-C kan denne erklæring nu skrives således:

```
alfa(int a, double b,
      double c, char *d)
```

Denne nye notation har den fordel at compileren laver typecheck og antalscheck på parametre, og den kan udføre nødvendige konverteringer af aktuelle parametre. Således vil følgende tre kald give kompileringfejle:

```
i = alfa(3, 4.2, 1.6, 2);
i = alfa(8);
```

```
i = alfa(3, 4.2, 1.6, "hej", 8);
```

Og i følgende kald vil 8-tallerne blive korrekt konverteret til doubles før de pushes på stakken:

```
i = alfa(3, 8, 8, "hej");
```

Den nye notation kræver imidlertid at rutinen erklæres før den bruges. Dette er i visse tilfælde umuligt, og der kan derfor blive tale om »forward«-erklæringer (à la Pascal), som det allerede nu er nødvendigt med rutiner der returnerer andet end integers. Hvis man ønsker at angive over for compileren hvordan *alfa* kaldes, sker det således:

```
alfa(int, double, double, char*);
```

Hvis *alfa* hører hjemme i et andet modul, bør nøgleordet *extern* benyttes.

ANSI-C forbyder (men *dcc* tillader) brugen af *typedef*-typenavne i sådanne rutineprototyper.

Hvis rutinen har et ukendt antal parametre af ukendte typer (som det fx. er tilfældet med *printf*) benyttes følgende notation (som dog af og til får *dcc* til at gå ned):

```
beta(int a, double b, ...);
```

Dette betyder at *beta* har mindst to parametre af de angivne typer, og derpå kommer et antal ukendte parametre.

Og så lige et par besynderligheder:

```
gamma();
```

Denne erklæring betyder *ikke* at *gamma* er en rutine med nul parametre, thi dette ville konflikte med gammel C. Erklæringen

betyder at gamma's parametre er ukendte. Her benytter man altså ikke gamma(...)!

```
delta(void);
```

Denne erklæring betyder at delta er en rutine med nul parametre. (Men også denne erklæring kan få gcc til at kløjs i det.)

By the way: Det er sund fornuft at erklære rutiner der ikke returnerer nogen værdi som værende af typen void:

```
void epsilon(int a)
```

1.2 Constants

Man kan nu erklære konstanter:

```
int const i = 8;
char const * p1;
char * const p2 = "hej";
char const * const p3 = "hejsa";
```

I disse erklæringer er i en konstant, dvs. compileren vil brokke sig hvis man forsøger at ændre i.

p1 er en pointer til en konstant, men p1 er ikke selv en konstant. Man kan med andre ord ændre p1 men ikke *p1.

p2 er en konstant pointer, men den peger ikke på en konstant. Man kan med andre ord ændre *p2 men ikke p2.

p3 er en konstant pointer til en konstant. Man kan med andre ord hverken ændre p3 eller *p3.

Brug af konstanter giver øget sikkerhed mod programmeringsfejl, og dette kan især være nyttigt i procedurkald: Lad os antage at jeg ønsker at kalde rutinen zeta med to parametre, et heltal og en tegnstring. Ingen af disse parametre må imidlertid ændres af rutinen. Antag at zeta erklæres således:

```
void zeta(int a, char *p)
{
    a=8;
    p[2]='x';
    zz(a,p);
}
```

Hvis jeg nu kalder zeta således:

```
int aa;
char *pp = "øllebrød";

zeta(aa,pp);
```

Vil jeg til min gru opleve at pp nu indeholder "øllebrød", fordi zeta er kodet galt. Hvis jeg i stedet havde erklæret zeta således:

```
void zeta(int a, char const *p)
{
    a=8;
    p[2]='x';
    zz(a,p);
}
```

ville jeg få en kompileringsfejl i tildelingen til p[2]. (Tildelingen til a er uskadelig.)

1.3 Volatile

Betrakt følgende program:

```
int x=0;

void hand()
{
    x=1;
}

main()
{
    int t=0;

    signal(SIGINT,hand);
    while (x==0) t++;
    printf("%d\n",t);
}
```

Meningen er at main skal tælle hvor længe der går inden brugeren får trykket på CTRL/C. Imidlertid vil en lidt for intelligent compiler her generere forkert kode. Den vil nemlig ræsonnere som følger: »Det eneste der sker inde i while-løkken er en optælling af t. Dette ændrer ikke værdien af x. Det er derfor spild af tid at teste værdien af x mere end én gang.« Compileren kan ikke vide at x ændres asynkront i hand, for måske ligger hand i en helt anden fil. Resultatet bliver at programmet ikke kommer til at virke efter hensigten.

Ved at ændre erklæringen af x til:

```
int volatile x=0;
```

fortæller man compileren at x kan ændre værdi bag om ryggen på programmet, og der derfor ikke må ske optimering af kode der rører x.

1.4 Tekststrengene

Man kan nu bruge hexadecimale værdier som tegn: Hvis man ønsker et *escape* (oktal værdi 033, hexadecimal værdi 0x1b) i et tegn har man hidtil været nødt til at skrive '\033'. Man kan nu også skrive '\x1b'.

Tekststrengene skrevet efter hinanden konkateneres. Sætningen

```
char *p = "alfa" "beta";
```

er ækvivalent med

```
char *p = "alfabeta";
```

Dette er specielt nyttigt ved tekststrengene som strækker sig over flere linier, eller det kan anvendes som følger:

```
#define HAK_START "\x05"
```

```
printf(HAK_START "X>Blank skærm\n");
```

1.5 void *

En ny type, pointer to void, er indført. Den betegner en pointer til hvad som helst. Med følgende erklæringer:

```
void *pv;
char *pc;
int *pi;
```

er følgende tildelinger lovlige:

```
pv=pc;
pc=pv;
pv=pi;
pi=pv;
```

mens følgende er forbudt:

```
pi=pc;
pc=pi;
```

ct [BP]

2. X/OPEN Security Guide

X/OPEN-gruppen har barslet med en »Security Guide«. Der er tale om en bog på ca. 130 sider. Den indeholder en lang række fornuftige råd og retningslinier om hvorledes man gør et UNIX-baseret styresystem mere sikkert. Det påstås ofte at UNIX ikke er et sikkert system, men meget af sårbarheden stammer fra at folk ikke er klar over hvorledes man konfigurerer og håndterer et sikkert system. Det er dette problem den nye Security Guide tager fat på. Der er altså

ikke tale om at den definerer nye faciliteter, kun at den beskriver den bedste udnyttelse af de eksisterende egenskaber.

En piratkopi af Security Guiden kan fås hos underegne.

ct [BP]

3. ISO-DIS 8859/9

Som bekendt er DDE's programmel baseret på benyttelse af det tegnsæt der er defineret i ISO standard 8859/1. Der er nu i ISO et nyt tegnsæt på vej som muligvis kan være interessant. Tegnsættet hedder 8859/9. Dette tegnsæt er identisk med 8859/1 med den ændring, at det understøtter tyrkisk i stedet for færøsk og islandsk.

I praksis betyder det, at de seks islandske (og til dels færøske) tegn, Ð, Ý, Þ, ð, ý og þ (i position D0, DD, DE, F0, FD og FE) erstattes med de tyrkiske tegn Ğ, İ, Ş, ğ, ı og ş, i den nævnte rækkefølge.

DDE bør muligvis overveje om 8859/9 dækker et vigtigere område end 8859/1.

ct [BP]

4. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes — brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse ct).

ct [BP]

VOVVOV

Nummer 12

3. maj 1989

I dette nummer:

Flytning af applikationssystemer fra ORACLE 5 til ORACLE 6
Ny prioriteringsalgoritme i SMOS
Write-bus på 68030
32-bits tegnsæt
Netnews
Opfordring

1. Flytning af applikationssystemer fra ORACLE 5 til ORACLE 6

Som alle ved er der en ny version af ORACLE på vej. Denne nye version 6 byder på en del ændringer/forbedringer i forhold til version 5.

Flere af ændringerne betyder at applikationssystemer skal ændres. Dette skrift beskæftiger sig med nogle af de ændringer der kan udføres under version 5 og vil virke under version 6. Dette indlæg har altså til formål at gøre overgangen til version 6 lettere.

Informationen er hentet på et ORACLE kursus, afholdt af en ORACLE medarbejder med stor indsigt i version 6.

Hvor det er nødvendigt vil jeg tillade mig at bruge noget pseudo SQL. Det er så op til læseren at oversætte det til HLI, Pro*COBOL, Pro*C eller SQL*Plus kommandofiler.

1.1 ROLLBACK på statement niveau

Version 6 laver fortrinsvis kun ROLLBACK på statement niveau. Dette betyder at programmer der for eksempel rammes af fejlen »duplicate value in index« selv skal sørge for en ROLLBACK, i modsætning til version 5 hvor hele transaktionen ruller tilbage hvorfor en COMMIT med henblik på at frigive låse normalt er udskadelig.

Eksempel: I version 5 vil følgende sekvens IKKE gemme data, mens der i version 6 vil blive gemt en række:

```
ON ERROR GOTO error;  
CREATE TABLE DEMOTAB(COL1 NUMBER);  
CREATE UNIQUE INDEX COL1IDX ON DEMOTAB(COL1);  
INSERT INTO DEMOTAB(COL1) VALUES(1);  
INSERT INTO DEMOTAB(COL1) VALUES(1); ❌ fejl
```

```
error:  
COMMIT
```

Hvis ovenstående skal virke på samme måde i version 6 som det er tilfældet i version 5 skal »error« håndteringen se således ud:

```
error:  
ROLLBACK  
COMMIT
```

1.2 FETCH over COMMIT

I forbindelse med indførelse ROW LEVEL locking er der ændret på hvornår rækker låses ved SELECT FOR UPDATE.

I version 5 sker låsning af en række når denne hentes med FETCH. I version 6 sker låsning af rækker når en cursor eksekveres (OPEN i Pro*C). Da COMMIT som bekendt frigiver låse er det altså ikke længere tilladt at FETCH'e efter COMMIT i version 6.

Ændringen kan lettest illustreres ved at vise hvilke låse der laves som funktion af de forskellige operationer. Forkortelserne ved låsene svarer til dem man kan se i »ods« på

version 5 og »monitor« i version 6:

Operation	Låsning i	
	version 5	version 6
parse	DDL(s)	TD(s) TD = Datadictionary lås
execute	DML(s)	TM(s) + ROW låse TM = Datamanipulation lås
fetch	ROW(x)	ingenting
commit		
fetch	ROW(x)	fejl-fetch across commit

Sekvenser med den ovenstående struktur vil ofte forekomme i iterationer hvor der ligger et stort select bag og hvor man selv har valgt at håndtere ROLLBACK og ganske enkelt COMMIT'er for fra tid til anden at lade andre komme til den tabel der opdateres.

Version 6's håndtering af låsning betyder iøvrigt at der er god grund til at være mest muligt restriktiv i WHERE delen på en SELECT FOR UPDATE.

1.3 Ændret layout i datadictionary

Datadictionary i version 6 er ændret således at der er en tabel (OBJ\$) der indeholder alle objekter. Objekter i version 6 er blandt andet tabeller, indexes og clustre. I version 5 er det muligt give disse tre objekter det samme navn, men det er ikke muligt i version 6.

Da det er min opfattelse at clustre er relativt lidt brugt vil det for de fleste være nok at omdøbe indexes der har samme navn som en tabel.

1.4 Afklaring vedr. SELECT FOR UPDATE OF col1, col2..... coln

Dette punkt er ikke medtaget fordi der skal ændres noget i relation til version 6, men blot for at fjerne enhver tvivl vedrørende hvilke kolonner der skal listets efter FOR UPDATE OF.

Det er fuldstændigt ligegyldigt hvilke kolonner man lister blot man husker at liste mindst en kolonne fra den tabel man vil opdatere på. Listningen af kolonner er nemlig kun med for at ORACLE kan vide i hvilken tabel der skal låses rækker i de tilfælde hvor der joines.

1.5 Omdøb objekter der har navnesammenfald med nøgleord/reserverede ord

Version 6 indholder en række nye nøgleord

der ikke bør bruges som kolonnenavne, tabelnavne, indexnavne og clusternavne. Når jeg skriver ikke bør, er det fordi de i fremtiden kan gå hen og blive til reserverede ord.

Jeg vil tillade mig kun at liste de mest sandsynlige:

BACKUP	LANGUAGE
BEGIN	LOGFILE
CHACHE	MINVALUE
CLOSE	MODULE
COBOL	MOUNT
CURSOR	NEXT
CYCLE	ONLY
DATAFILE	OPEN
DOUBLE	PASCAL
END	READ
ESCAPE	REAL
EXEC	RELEASE
GO	
GOTO	
INCLUDING	
INDICATOR	
INT	
KEY	

1.6 Ændring i sorteringsalgoritmen

I version 5 er NULL den mindste værdi og i version 6 den største værdi. Det er altså værd lige at kigge lidt på ORDER BY og se om der sorteres på kolonner der kan være NULL. Vil man tage højde for ændringen allerede nu er løsningen at sortere på NVL(COL1, ' ').

Det skal endnu engang kraftigt pointeres at dette skrift kun indeholder opgaver der kan udføres på forhånd og ikke fra skal opfattes som en fuldstændig oversigt. Intentionen er ganske enkelt at det der er rettet ikke skal rettes senere og flere af de ovenstående punkter betyder det jo ikke noget lige at have i baghovedet under udvikling eller tilretning af et produkt.

I forbindelse med et ORACLE 6 seminar den 17 maj kl. 13.00 til 16.00 i Herlev, vil der blive givet en bredere orientering. Seminaret vil blive annonceret i Super-Posten.

tk [AS]

2. Ny prioriteringsalgoritme i SMOS

Med fremtidige frigivelser af SMOS vil der være indarbejdet en ny måde at prioritere processer på.

Hidtil har vi kørt med en scheduling, der kort sagt fungerer som følger:

På hver MCU haves en liste af aktive processer, dvs processer, der er klar til at køre, hvis de ellers kunne få tildelt CPU-tid. Når der skal scheduleres (hvad enten det drejer sig om, at kørende proces suspenderes eller, der kommer et timerinterrupt), vælges blandt de processer i køen, der har bedst prioritet, den der står forrest. Da indsættelse i køen sker bagi udvælges på denne måde den af de bedst prioriterede processer, der har været længst i køen.

En proces har den samme prioritet i det meste af sin levetid. Denne prioritet er den samme, som den nice-værdi processen fødes med. (Denne kan selvfølgelig ændres ved systemkaldet »nice« eller utiliten »nice«, hvorved prioriteten også ændres). I få situationer ændres prioriteten midlertidigt; en proces, der er suspenderet, fordi den venter på interrupt fra en I/O enhed, vil ved indsættelse i køen over aktive processer, når interruptet er kommet, få justeret sin prioritet, så den bliver en anelse bedre. Næste gang processen scheduleres ud, bliver prioriteten sat tilbage til den oprindelige værdi igen.

Ovennævnte er den eneste form for prioritetsmodning, der finder sted i SMOS nu.

HUSK: jo mindre et tal desto bedre prioritet.

2.1 Den nye scheduling

Baggrunden for at indføre en anden form for scheduling er bl.a. at undgå visse fejlsituationer, der er set med den gamle scheduling. Det er en ny form for scheduling set fra vores side, men er rent faktisk den, der anvendes i AT&T's UNIX system V.

Den nye måde at scheduleres på afviger fra den gamle på to væsentlige punkter:

- prioritetsmodning
- opdeling af prioriteter i bruger- og systemprioriteter

Ved prioritetsmodning forstås, at en proces, der har brugt meget CPU-tid, midlertidigt får sat sin prioritet ringere, men denne modnes igen til den oprindelige bedre prioritet, efterhånden som andre processer har fået CPU-tid.

En proces' prioritet ligger i de gamle systemer i intervallet mellem -20 og 20. I den nye scheduling anvendes et interval fra 0 til 80. Opdelingen i bruger- og systemprioriteter er skjult for brugeren, der stadigvæk »ser« 41 prioriteter mellem -20 og 20, men internt »skubbes« disse prioriteter til at ligge fra (i første omgang) 20 til 60. Ved ovennævnte prioritetsmodning kan hele det interne interval af brugerprioriteter komme i anvendelse — dette går fra 20 til 80. Af det samlede prioritetsinterval er de første 20 prioriteter forbeholdt processer, der kører i supervisor-mode. Disse prioriteter tildeles (og fjernes igen) af styresystemet. Disse tildeles processer, der suspenderer sig selv, f.eks. fordi de venter på I/O.

Med den nye scheduling opnås to ønskelige faciliteter:

1. En proces i supervisor-mode kan aldrig blive holdt væk fra at køre af en proces i user-mode. Herved undgås de »låste« maskiner, vi sommetider har set, hvor et højtprioriteret program, fordi det kører i uendelig løkke, forhindrer et andet program, der p.t. har reserveret en global ressource, i at komme til, hvorved hele maskinen efterhånden går i stå på nær den uendelige løkke.
2. Programmer, der kører i uendelig løkke, vil i det hele taget komme sjældnere til at køre, da de vil få deres prioritet sat ned som følge af prioritetsmodningen; men de vil selvfølgelig køre konstant, hvis der ikke er andre aktive processer på pågældende CPU.

abc [BP]

3. Write-bus på 68030

En ny måde at sende data mellem CPU'erne i en Supermax.

Enhederne på fællesbussen i en Supermax kan sammenlignes med en skoleklasse, hvor hver elev har sit kladdehæfte. Eleverne kan uforstyrret skrive og læse i hvert sit kladdehæfte; men de har også adgang til at skrive og læse i de andre elevers hæfter.

Proceduren for dette er imidlertid lidt omstændelig (eftersom eleverne er meget glemsomme og aldrig kan huske hinanden):

Hvis **Claus** vil **skrive** noget i **Bos** kladdehæfte, beder Claus om ordet, hvorefter han med høj røst udråber: »Bo, skriv et Y på side 17 linie 8!«. Alle eleverne hører udråbet, men kun Bo vil udføre kommandoen og derefter svare »Udført!«.

Hvis **Ole** vil **læse** i **Bos** kladdehæfte, beder Ole om ordet og udråber: »Bo, hvad står der på side 15 linie 6?!«. Alle eleverne hører budskabet, men kun Bo vil reagere ved at svare.

Lad os sige, at Bo er udnævnt til klassens duks (sådan en skal der altid være i en klasse), som har de ting i sit hæfte, alle eleverne hele tiden skal enes om. Det giver en masse råben og forstyrrelse, når eleverne ustandselig skal spørge ordensduksen om information.

Med 68030 har eleverne lært noget nyt: Når man skal **skrive** noget, kan man ringe med en klokke i stedet for at sige et navn. Så vil **alle** eleverne udføre skrivningen i deres hæfter og svare »Forstået!«.

Den nye lærdom er udnyttet på den måde, at eleverne er blevet udstyret med hver sin kopi af duksens ordenshæfte, som de kan læse uden at blive forsinket og uden at forstyrre de andre. Til gengæld er det nødvendigt, at eleverne altid benytter klokken, når de vil skrive i ordenshæftet (det gælder også duksen, som før i tiden kunne skrive i ro og mag hos sig selv).

På fællesbussen i en Supermax sker der noget lignende.
CPU'erne er klassens elever.
Enhedsnumrene er elevernes navne.
Master-CPU'en er klassens duks.

Fælleslageret er duksens ordenshæfte. Busadressen indeholder altid et enhedsnummer.

Den nye »klokke« er en bestemt bit i adressen.

Lærdommen træder i kraft med SMOS version 530.1J.

Hvad er det så, vi ser:

Fælleslageret er kopieret ud til alle CPU'er. Læsning i fælleslageret bliver altid lokale læsninger. Skrivninger i fælleslageret er altid broadcasts på bussen.

3.1 Tidsmålinger

Målt med multishell bliver master-CPU'en 5% langsommere, mens de øvrige CPU'er bliver 13% hurtigere! Den mindskede aktivitet på bussen vil give yderligere forbedringer på store maskiner.

3.2 Konkurrenter

Så vidt vi tror i BP, er der ikke andre fler-CPU maskiner, der kobler lager sammen på denne nye og forholdsvis enkle måde. I USA nærmer man sig en lignende egenskab ved hjælp af meget kompliceret og dyr cache-elektronik.

es [BP]

4. 32-bits tegnsæt

Indtil videre har de kun akademisk interesse, men alligevel: Nu hvor vi lige har vænnet os til at bruge et 8-bits tegnsæt er ISO på vej med et 32-bits tegnsæt. Det foreligger kun på skitseplan endnu, men idéen er interessant, og jeg vil derfor her give en kort gennemgang af tegnsættet.

Meningen med tegnsættet er at det skal dække *alle verdens sprog!* Intet mindre. Det indeholder både latinske tegn, russiske tegn, græske tegn, kinesiske tegn, japanske tegn osv. osv. Der er tegn fra mange alfabeter jeg aldrig har hørt tale om før, såsom devanagari, gurmukhi, gujerati, oriya, og meget andet.

Hvert tegn kodes altså med 4 bytes. For printbare tegn har hver eneste af disse bytes en værdi mellem 0x20 og 0x7f eller 0xa0 og 0xff, altså de tegn som normalt i vort eget gængse 8-bits tegnsæt er printbare.

32 bits er betydeligt mere end man har brug for. Faktisk er standarden endnu ikke nået længere end til at benytte 16 af de 32 bits, men det er rart at være forberedt på fremtiden. De fire bytes i tegnkoden skal fortolkes som følger:

- Den første byte i tegnkoden udpeger en såkaldt »gruppe« i tegnsættet. Denne byte har altid værdien 0x20, man har simpelthen ikke brug for at ændre på den.
- Den anden byte udpeger en såkaldt »plan« inden for tegnsætsgruppen. En plan består altså af 256 x 256 tegnkoder. Denne byte er indtil videre altid 0x20, men man har planer om udvidelser her: Kode 0x21 til 0xdf er reserveret til fremtidige udvidelser af tegnsættet. Kode 0xe0 til 0xff er til fri brug for implementører.
- Den tredje byte udpeger en såkaldt »række« inden for tegnsætplanen. En sådan række er altså et 8-bit tegnsæt. Række 0x20 er vort kendte ISO 8859/1-tegnsæt, som benyttes på Supermax.
- Den fjerde byte udpeger det enkelte tegn inden for tegnsætrækken.

Så hvis I fx. har brug for det kendte tibetanske tegn »chuchenyige«, skal I vide at det har kode 0x2020a365. Et almindeligt dansk »Æ« har kode 0x202020c6.

Det nye tegnsæt giver mulighed for at man kan nøjes med at anvende delmængder af det. Det kan fx. benyttes i en 16-bits eller 8-bits udgave, hvor man starter med at specificere hvilken plan eller række af tegnsættet man ønsker at benytte. Faktisk er default-tilstanden den, at man benytter en 8-bits delmængde af tegnsættet, som faktisk er række 0x20 af plan 0x20 af gruppe 0x20, altså vort eget 8-bits tegnsæt.

Udkastet til tegnsættet, som indtil videre kendes under navnet ISO DP 10646 (DP betyder »Draft Proposal«), kan ses hos mig.

ct [BP]

5. Netnews

Vi har nu haft netnews kørende i halvandet år i DDE. Jeg vil derfor benytte lejligheden

til at komme med en frisk liste over tilgængelige nyhedsgrupper, samt en oversigt over hvilke nyhedsgrupper der modtages af hvilke maskiner i DDE. Denne liste er vedlagt dette nummer af Vovov.

ct [BP]

6. Opfordring

Vovov er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes – brug fx en diskette, eller send mig artiklen med UUCP-mail (postadresse *ct*).

ct [BP]

Liste over nyhedsgrupper

Det følgende er en liste over alle de nyhedsgrupper DDE modtager via netnews. Listen stammer fra medio april, og der er sket nogle tilføjelser siden da (men jeg magtede ikke at opdatere den – det er en rimelig horribel opgave). Første kolonne er navnet på nyhedsgruppen, anden kolonne beskriver hvilke maskiner der modtager den pågældende gruppe, og tredje kolonne er en kort beskrivelse af gruppen.

Maskinerne er forkortet således: H - HAL9000, K - Kurt, c - caspian, i - ipIV, n - nessie, t - tsaren. Maskinerne mother og Aragorn er ikke nævnt nedenfor, idet disse to maskiner modtager samtlige nyhedsgrupper.

alt.gourmand	Hcn	Recipes & cooking info.
alt.hypertext	c	Discussion of hypertext -- uses, transport, etc.
alt.sources	Kcnt	Alternative source code, unmoderated. Caveat Emptor.
alt.sources.amiga		Technically oriented Amiga PC sources.
comp.ai	t	Artificial intelligence discussions.
comp.ai.digest	t	Artificial Intelligence discussions.
comp.ai.edu	t	Applications of Artificial Intelligence to Education.
comp.ai.neural-nets	t	All aspects of neural networks.
comp.ai.nlang-know-rep	t	Natural Language and Knowledge Representation.
comp.ai.vision	t	Artificial Intelligence Vision Research.
comp.arch	c	Computer architecture.
comp.archives	c	Archives of public domain software.
comp.binaries.amiga	H	Encoded public domain programs in binary.
comp.binaries.apple2	K	Binary-only postings for the Apple II computer.
comp.binaries.atari.st		Binary-only postings for the Atari ST.
comp.binaries.hypercard		Binary-only postings of Macintosh HyperCard stacks.
comp.binaries.ibm.pc	cnt	Binary-only postings for IBM PC/MS-DOS.
comp.binaries.ibm.pc.d	Hct	Discussions about IBM/PC binary postings.
comp.binaries.mac	K	Encoded Macintosh programs in binary.
comp.bugs.2bsd		Reports of UNIX* version 2BSD related bugs.
comp.bugs.4bsd		Reports of UNIX version 4BSD related bugs.
comp.bugs.4bsd.ucb-fixes		Bug reports/fixes for BSD Unix.
comp.bugs.misc	Kct	General bug reports and fixes (includes V7 & uucp).
comp.bugs.sys5	Kct	Reports of USG (System III, V, etc.) bugs.
comp.cog-eng	t	Cognitive engineering.
comp.compilers	ct	Compiler construction, theory, etc.
comp.databases	t	Database and data management issues and theory.
comp.dcom.lans	Knt	Local area network hardware and software.
comp.dcom.lans.hyperchannel		Hyperchannel networks within a IP network.
comp.dcom.lans.v2lni		Proteon Pronet/V2LNI Ring networks.
comp.dcom.modems	t	Data communications hardware and software.
comp.dcom.telecom	t	Telecommunications digest.
comp.doc	cnt	Archived public-domain documentation.
comp.doc.techreports	cnt	Lists of technical reports.
comp.editors	ct	Topics related to computerized text editing.
comp.edu	ct	Computer science education.
comp.edu.composition	ct	Writing instruction in computer-based classrooms.
comp.emacs	Knt	EMACS editors of different flavors.
comp.fonts	c	Computer typefonts -- design, conversion, use, etc.
comp.graphics	cint	Computer graphics, art, animation, image processing.
comp.graphics.digest	cint	Graphics software, hardware, theory, etc.

comp.ivideodisc	t	Discussion about interactive video discs
comp.lang.ada	t	Discussion about Ada*.
comp.lang.apl	t	Discussion about APL.
comp.lang.c	Kcit	Discussion about C.
comp.lang.c++	Kcit	The object-oriented C++ language.
comp.lang.clu	t	The CLU language & related topics.
comp.lang.eiffel	c	The eiffel programming language.
comp.lang.forth	t	Discussion about Forth.
comp.lang.forth.mac	t	The CSI MacForth programming environment.
comp.lang.fortran	t	Discussion about FORTRAN.
comp.lang.icon	t	Topics related to the ICON programming language.
comp.lang.idl	t	IDL (Interface Description Language) related topics.
comp.lang.lisp	t	Discussion about LISP.
comp.lang.lisp.franz	t	The Franz Lisp programming language.
comp.lang.lisp.x	t	The XLISP language system.
comp.lang.misc	ct	Different computer languages not specifically listed.
comp.lang.modula2	ct	Discussion about Modula-2.
comp.lang.pascal	ct	Discussion about Pascal.
comp.lang.postscript	ct	The PostScript Page Description Language.
comp.lang.prolog	t	Discussion about PROLOG.
comp.lang.rexx	t	The REXX command language.
comp.lang.scheme	t	The Scheme Programming language.
comp.lang.scheme.c	t	The Scheme language environment.
comp.lang.sigplan	c	?????
comp.lang.smalltalk	ct	Discussion about Smalltalk 80.
comp.lang.visual	t	Visual programming languages.
comp.laser-printers	Hcnt	Laser printers, hardware & software.
comp.lsi	nt	Large scale integrated circuits.
comp.lsi.cad	int	Electrical Computer Aided Design.
comp.mail.elm	Ht	Discussion and fixes for the ELM mail system.
comp.mail.headers	Ht	Gatewayed from the ARPA header-people list.
comp.mail.mh	Ht	The UCI version of the Rand Message Handling system.
comp.mail.misc	Ht	General discussions about computer mail.
comp.mail.multi-media	Hct	Multimedia Mail.
comp.mail.sendmail	H	Discussion of the Berkeley sendmail mailer.
comp.mail.uucp	HKt	Mail in the uucp network environment.
comp.misc	crt	General topics about computers not covered elsewhere.
comp.music	Hc	Applications of computers in music research.
comp.newprod	ct	Announcements of new products of interest.
comp.org.decus	t	DEC* Users' Society newsgroup.
comp.org.fidonet	t	FidoNews digest, official news of FidoNet Assoc.
comp.org.ieee	c	The IEEE organization.
comp.org.userix	ct	USENIX Association events and announcements.
comp.org.usrgroup	c	News and discussion about/from the /usr/group organization.
comp.os.aos		Topics related to Data General's AOS/VS.
comp.os.cpm		Discussion about the CP/M operating system.
comp.os.cpm.amethyst		Discussion of Amethyst, CP/M-80 software packages.
comp.os.eunice		The SRI Eunice system.
comp.os.minix	ct	Discussion of Tanenbaum's MINIX system.
comp.os.misc	ct	General OS-oriented discussion not carried elsewhere.
comp.os.os9		Discussions about the os9 operating system.
comp.os.research	c	Operating systems and related areas.
comp.os.rsts		Topics related to the PDP-11 RSTS/E operating system.

comp.os.v	t	The V distributed operating system from Stanford.
comp.os.vms		DEC's VAX* line of computers & VMS.
comp.os.xinu	t	The XINU operating system from Purdue (D. Comer).
comp.parallel	c	Massively parallel systems: hardware & software.
comp.periphs	Hn	Peripheral devices.
comp.periphs.printers	Ht	Information on printers.
comp.protocols.appletalk	K	Applebus hardware & software.
comp.protocols.ibm	K	Networking with IBM mainframes.
comp.protocols.iso	Kt	The ISO protocol stack.
comp.protocols.iso.dev-environ	t	The ISO Development Environment.
comp.protocols.iso.x400		X400 mail protocol discussions.
comp.protocols.iso.x400.gateway		X400 mail gateway discussions.
comp.protocols.kerberos		?????
comp.protocols.kermit	HKt	Info about the Kermit package.
comp.protocols.misc	Kt	Various forms and types of FTP protocol.
comp.protocols.nfs		Discussion about the Network File System protocol.
comp.protocols.pcnet	HKt	Topics related to PCNET (a personal computer network).
comp.protocols.pup	t	The Xerox PUP network protocols.
comp.protocols.tcp-ip	HKt	TCP and IP network protocols.
comp.protocols.tcp-ip.domains	HKt	Topics related to Domain Style names.
comp.protocols.tcp-ip.ibmpc	HKt	TCP/IP for IBM(-like) personal computers.
comp.risks	cnt	Risks to the public from computers & users.
comp.simulation		Simulation methods, problems, uses.
comp.society	cnt	The impact of technology on society.
comp.society.futures	t	Events in technology affecting future computing.
comp.society.women	c	Women's roles and problems in computing.
comp.software-eng	ct	Software Engineering and related topics.
comp.sources.amiga	Hi	Source code-only postings for the Amiga.
comp.sources.atari.st	i	Source code-only postings for the Atari ST.
comp.sources.bugs	Kcint	For bug fixes and features discussion
comp.sources.d	HKcint	For any discussion of source postings.
comp.sources.games	HKcint	Postings of recreational software.
comp.sources.games.bugs	Hci	Recreational software: bugs and fixes.
comp.sources.mac	Ki	Software for the Apple Macintosh.
comp.sources.misc	Kcint	Posting of software.
comp.sources.sun	i	Software for the Sun computers.
comp.sources.unix	Kcint	Postings of public-domain sources.
comp.sources.wanted	Hcit	Requests for software and fixes.
comp.sources.x	cit	Software for the X windows system.
comp.std.c	ct	Discussion about C language standards.
comp.std.internat	nt	Discussion about international standards.
comp.std.misc	nt	Discussion about various standards.
comp.std.mumps		Discussion for the X11.1 committee on Mumps.
comp.std.unix	Hcnt	Discussion for the P1003 committee on UNIX.
comp.sys.amiga	H	Commodore Amiga: info&uses, but no programs.
comp.sys.amiga.tech	H	Technical discussion about the Amiga.
comp.sys.apollo	n	Apollo computer systems.
comp.sys.apple	K	Discussion about Apple micros.
comp.sys.atari.8bit		Discussion about 8 bit Atari micros.
comp.sys.atari.st		Discussion about 16 bit Atari micros.
comp.sys.att	Kt	Discussions about AT&T microcomputers.
comp.sys.cbm		Discussion about Commodore micros.
comp.sys.cdc		Control Data Corporation Computers (e.g., Cybers).

comp.sys.celerity		Celerity Computers.
comp.sys.dec		Discussions about DEC computer systems.
comp.sys.dec.micro		DEC Micros (Rainbow, Professional 350/380).
comp.sys.encore		Encore's MultiMax computers.
comp.sys.handhelds	c	Handheld computers and programmable calculators.
comp.sys.hp	n	Discussion about Hewlett-Packard equipment.
comp.sys.ibm.pc	Hcnt	Discussion about IBM personal computers.
comp.sys.ibm.pc.digest	cnt	The IBM PC, PC-XT, and PC-AT.
comp.sys.ibm.pc.rt	t	Topics related to IBM's RT computer.
comp.sys.intel	c	Discussions about Intel systems and parts.
comp.sys.intel.ipsc310		Anything related to Xenix on an Intel 310.
comp.sys.m6809		Discussion about 6809's.
comp.sys.m68k	c	Discussion about 68k's.
comp.sys.m68k.pc	n	Discussion about 68k-based PCs.
comp.sys.mac	K	Discussions about the Apple Macintosh & Lisa.
comp.sys.mac.digest	K	Apple Macintosh: info&uses, but no programs.
comp.sys.mac.hypercard	K	The Macintosh Hypercard: info & uses.
comp.sys.mac.programmer	K	Discussion by people programming the Apple Macintosh.
comp.sys.masscomp		The Masscomp line of computers.
comp.sys.misc	cnt	Discussion about computers of all kinds.
comp.sys.next	c	Discussion about the new NeXT computer.
comp.sys.northstar		Northstar microcomputer users.
comp.sys.nsc.32k		National Semiconductor 32000 series chips.
comp.sys.proteon		Proteon gateway products.
comp.sys.pyramid		Pyramid 90x computers.
comp.sys.ridge		Ridge 32 computers and ROS.
comp.sys.sequent		Sequent systems, (Balance and Symmetry).
comp.sys.sgi		Silicon Graphics' Iris workstations and software.
comp.sys.sun	n	Sun "workstation" computers.
comp.sys.super		Supercomputers.
comp.sys.tahoe		The CCI 6.32, Harris HCX/7, Sperry 7000 computer series.
comp.sys.tandy		Discussion about TRS-80's.
comp.sys.ti		Discussion about Texas Instruments.
comp.sys.ti.explorer		The Texas Instruments Explorer.
comp.sys.transputer	n	The Transputer computer and OCCAM language.
comp.sys.workstations	n	Various workstation-type computers.
comp.sys.xerox		Xerox 100 workstations and protocols.
comp.sys.zenith		Heath terminals and related Zenith products.
comp.sys.zenith.z100		The Zenith Z-100 (Heath H-100) family of computers.
comp.terminals	Hnt	All sorts of terminals.
comp.terminals.bitgraph	nt	The BB&N BitGraph Terminal.
comp.terminals.tty5620		AT&T Dot Mapped Display Terminals (5620 and BLIT).
comp.text	cnt	Text processing issues and methods.
comp.text.desktop	cnt	Technology & techniques of desktop publishing.
comp.theory	ct	Theoretical Computer Science.
comp.theory.cell-automata	ct	Discussion of all aspects of cellular automata.
comp.theory.dynamic-sys	t	Ergodic Theory and Dynamical Systems.
comp.theory.info-retrieval	Ht	Topics related to Information Retrieval.
comp.theory.self-org-sys	t	Topics related to self-organization.
comp.unix	HKcnt	Discussion of UNIX* features and bugs.
comp.unix.aux		Discussion of MAC II A/UX UNIX.
comp.unix.cray	n	Discussion about Cray UNIX.
comp.unix.i386	c	Discussion about 386-UNIX.

comp.unix.microport		Discussion of Microport's UNIX.
comp.unix.questions	HKcnt	UNIX neophytes group.
comp.unix.ultrix		Discussions about DEC's Ultrix.
comp.unix.wizards	HKcnt	Discussions, bug reports, and fixes on and for UNIX.
comp.unix.xenix	n	Discussion about the Xenix OS.
comp.windows.misc	cint	Various issues about windowing systems.
comp.windows.ms	i	Window systems under MS/DOS.
comp.windows.news	cint	Sun Microsystems' News window system.
comp.windows.x	cint	Discussion about the X Window System.
dde.general	HKcint	Generelle DDE-diskussioner.
dde.kalender	HKcit	Diskussioner om DDE's kalendersystem.
dde.kartotek	HKcit	Diskussioner om DDE's kartotekssystem.
dde.kontor	HKcit	Diskussioner om DDE's kontorsystem.
dde.oracle4	Kcit	Diskussioner om Oracle 4.
dde.oracle5	HKcit	Diskussioner om Oracle 5.
dde.post	HKcit	Diskussioner om DDE's postsystem.
dde.programs	HKcint	Distribution af DDE programmer.
dde.regneark	HKcit	Diskussioner om DDE's regneark.
dde.sag	HKcit	Diskussioner om DDE's sagssystem.
dde.service	HKcint	Diskussioner om DDE's serviceafdeling.
dde.swnotes	HKcint	Software notes.
dde.tekst	HKcit	Diskussioner om DDE's tekstbehandlingsystem.
dde.test	HKcint	Test.
dde.unix	HKcint	Diskussioner om DDE's styresystem.
dde.vinduer	HKcit	Diskussioner om DDE's vinduessystem.
dk.foredrag	ci	Foredrag i Danmark.
dk.general	HKcint	General information for danskere.
dk.jobs	HKcit	Danske jobannoncer.
dk.test	Hcit	Test.
eunet.bugs.4bsd		Bug reports on 4.*bsd, limited (EUnet) "audience".
eunet.bugs.uucp	HKt	Bug reports on uucp, specifically EUUG uucp.
eunet.esprit	ct	Discussion/info about Esprit projects.
eunet.followup	Hct	Followup group for eunet.general.
eunet.general	Hct	General discussions.
eunet.jokes	HKcit	Jokes only Europeans can (do?) understand... :-)
eunet.micro.acorn		Discussion about Acorn micros.
eunet.newprod	c	Announcements of new products of European interest.
eunet.news	Hcnt	Group for news topics, limited to EUnet.
eunet.news.group	Hcnt	Discussion on and proposals for new eunet newsgroups.
eunet.politics		(European) political discussions (and flames!).
eunet.sources	Kcint	EUnet-wide (only!) group for posting sources.
eunet.test	H	EUnet-wide test group.
eunet.works	cint	Workstations (specifically European) topics.
misc.jobs.misc	Hnt	Discussion about employment, workplaces, careers.
misc.jobs.offered	Hnt	Announcements of positions available.
misc.jobs.resumes		Postings of resumes and "situation wanted" articles.
misc.misc	Kct	Various discussions not fitting in any other group.
misc.security	ct	Security in general, not necessarily computer related.
misc.wanted	Ht	Requests for things that are needed (NOT software).
news.admin	Ht	Comments directed to news administrators.
news.announce.conferences	Hct	Calls for papers and conference announcements.
news.announce.important	HKcnt	General announcements of interest to all.
news.announce.newusers	Hcnt	Explanatory postings for new users.

news.config	Ht	Postings of system down times and interruptions.
news.groups	Hcnt	Discussions and lists of newsgroups.
news.lists	Hcnt	News-related statistics and lists.
news.misc	Hct	Discussions of USENET itself.
news.software.b	HKt	Discussion about B news software.
news.software.nntp	t	The Network News Transfer Protocol.
news.software.notes	HKt	Notesfile software from the Univ. of Illinois.
news.sysadmin	Hcnt	Comments directed to system administrators.
rec.arts.books		Books of all genres, and the publishing industry.
rec.arts.movies		Discussions of movies and movie making.
rec.arts.sf-lovers	t	Science fiction lovers' newsgroup.
rec.games.board	it	Discussion and hints on board games.
rec.games.bridge	it	Hobbyists interested in bridge.
rec.games.chess	it	Chess & computer chess.
rec.games.empire	it	Discussion and hints about Empire.
rec.games.frp	it	Discussion about Fantasy Role Playing games.
rec.games.go	it	Discussion about Go.
rec.games.hack	HKIt	Discussion, hints, etc. about the Hack game.
rec.games.misc	it	Games and computer games.
rec.games.moria	i	Comments, hints, and info about the Moria game.
rec.games.pbm	it	Discussion about Play by Mail games.
rec.games.programmer	Hit	Discussion of adventure game programming.
rec.games.rogue	it	Discussion and hints about Rogue.
rec.games.trivia	it	Discussion about trivia.
rec.games.vectrex	it	The Vectrex game system.
rec.games.video	it	Discussion about video games.
rec.ham-radio		Amateur Radio practices, contests, events, rules, etc.
rec.ham-radio.packet		Discussion about packet radio setups.
rec.ham-radio.swap-d		????
rec.humor.funny		Jokes that are funny (in the moderator's opinion).
rec.mag	c	Magazine summaries, tables of contents, etc.
rec.mag.fsfnet	c	A Science Fiction "fanzine".
rec.mag.otherrealms	c	Edited science fiction & fantasy "magazine".
rec.music.beatles		Postings about the Fab Four & their music.
rec.music.bluenote		Discussion of jazz, blues, and related types of music.
rec.music.cd		Music on Compact Discs.
rec.music.classical		Discussion about classical music.
rec.music.dementia		Crazy music.
rec.music.folk		Folks discussing folk music of various sorts.
rec.music.gaffa		Progressive music (e.g., Kate Bush).
rec.music.gdead		A group for (Grateful) Dead-heads.
rec.music.makers	i	For performers and their discussions.
rec.music.misc		Music lovers' group.
rec.music.synth	i	Synthesizers and computer music.
sci.astro	c	Astronomy discussions and information.
sci.bio		Biology and related sciences.
sci.bio.technology	i	Any topic related to biotechnology.
sci.chem		Chemistry.
sci.crypt	t	Different methods of data en/decryption.
sci.edu		Education.
sci.electronics	nt	Circuits, theory, electrons and discussions.
sci.environment		Environmental discussions.
sci.lang		Natural languages, communication, etc.

sci.lang.japan		The Japanese language, both spoken and written.
sci.logic		Logic – math, philosophy & computational aspects.
sci.math	ct	Mathematical discussions and pursuits.
sci.math.num-analysis	ct	Numerical Analysis.
sci.math.stat	ct	Statistics discussion.
sci.math.symbolic	ct	Symbolic algebra discussion.
sci.med		Medicine and its related products and regulations.
sci.med.aids		Medicine subgroup for AIDS related topics.
sci.military		Discussion about science and the military.
sci.misc	ct	Short-lived discussions on subjects in the sciences.
sci.nanotech	c	Self-reproducing molecular-scale machines.
sci.philosophy.meta		Discussions within the scope of "MetaPhilosophy".
sci.philosophy.tech		Technical philosophy: math, science, logic, etc.
sci.physics	ci	Physical laws, properties, etc.
sci.psychology		Topics related to psychology.
sci.research	c	Research methods, funding, ethics, and whatever.
sci.space	ct	Space, space programs, space related research, etc.
sci.space.shuttle	ct	The space shuttle and the STS program.
soc.culture.african		Discussions about Africa & things African.
soc.culture.arabic		Technological & cultural issues, *not* politics.
soc.culture.celtic		Group about Celtics (*not* basketball!).
soc.culture.china		About China and Chinese culture.
soc.culture.esperanto		The neutral international language Esperanto.
soc.culture.greek		Group about Greeks.
soc.culture.indian		Group for discussion about India & things Indian.
soc.culture.japan		Everything Japanese, except the Japanese language.
soc.culture.jewish	c	Group for discussion about Jewish culture & religion.
soc.culture.misc		Group for discussion about other cultures.
soc.human-nets		Computer aided communications digest.
soc.politics		Political problems, systems, solutions.
soc.politics.arms-d		Arms discussion digest.

VOVVOV

Nummer 15

5. marts 1991

I dette nummer:

Så gør hunden igen
C-udvikling på heterogen Supermax
Kodegenbrug i C++
Indtryk fra Look91 Konferencen
CSCW
Opfordring

1. Så gør hunden igen

Efter en pause på mere end et år udkommer Vovov nu atter. Redaktøren har tilbragt hele 1990 med fødderne opad, men er nu atter blevet inverteret, og derfor skulle en mere regelmæssig gøen i den kommende tid være mulig.

[Accenten over ø i overskriften er helt korrekt efter den nye retskrivningsordbog – med et pennestrøg har man pludselig indført et nyt tegn i dansk, som ikke findes i ISO-8859/1 tegnsættet.]

ct [BP]

2. C-udvikling på heterogen Supermax

2.1 Introduktion

Formålet med denne artikel er at klarlægge de problemer, der er med et programmeludviklingsmiljø på en heterogen Supermax, og hvordan de er løst.

2.2 Baggrund

Der kan være 2 typer CPU'er i en heterogen Supermax:

- CISC-CPU'er, Motorola M68030.
- RISC-CPU'er, MIPS R3000.

De to typer CPU'er har bl.a. helt forskellige instruktionssæt, og kan derfor *ikke* afvikle »hinandens« programmer. Der er et krav om

at kunne afvikle binære MIPS-programmer på R3000-CPU'en, dvs programmer som end ikke skal linkes på en Supermax.

De »rene« 68030- og 68020-Supermax'er er specialtilfælde af den heterogene Supermax, og er SWD-mæssigt magen til. (SWD: SoftWare Development ~ Programmel Udviklingspakke).

2.3 Problemer

Nedenfor gennemgås følgende problemer og den måde de er løst på:

- Hvis der både skal være en RISC- og en CISC-udgave af et program, hvordan skal de så ligge; f.eks. hvad skal `/bin/sh` være? (HEM-filer)
- Generering af programmer.
 - Datakompatibilitet. (Kan generelt ikke opnås mellem MIPS-binære programmer og DDE-Motorola programmer). (MI/MO-problematik)
 - Hvordan bestemmes det hvad programmet genereres til? (TARGETMC)

2.4 HEM-filer.

For at kunne balancere en heterogen Supermax, er det ønskeligt at en række programmer findes i 2 udgaver, så de kan udføres på begge typer processorer. Men samtidig ligger navnet på mange programmer gemt som absolutte filnavne, så – voilà – vi lægger begge typer kode i en og samme fil, i det følgende kaldet en HEM-fil. (HEM:

Heterogeneous Executable Module).

Når operativ-systemet skal udføre et program, der ligger som HEM-fil, udføres den kode, som svarer til den mindst belastede CPU.

Opbygningen af en HEM-fil er i princippet enkel: Forrest et hoved, med information om f.eks.:

- Et HEM-magic-nummer.
- Starten på Motorola-delen (hvis den er med).
- Starten på R3000-delen (hvis den er med).
- Diverse tuningsparametre, som ikke skal omtales her.

Herefter ligger den »egentlige« kode for hver del.

2.5 Datakompatibilitet og TARGETMC

Udlægning af datastrukturer er *ikke* ens i vores gamle Motorola-verden og den nye MIPS verden.

Forskelle i data-udlæg har betydning for programmer, som udveksler data med hinanden via f.eks. pipes, filer og fælles lager.

Dette er et problem, da vi som før omtalt ønsker at anvende binære programmer fra andre »MIPS-dåser«.

Vi skal altså kunne generere 2 slags kode og 2 slags data:

- M68020/M68030-kode og R3000-kode og
- Motorola-udlæg og MIPS-udlæg af strukturer.

I alt bliver det til 3 muligheder, da vi *ikke* vil understøtte Motorola-kode med MIPS-data-udlæg.

Environmentet **TARGETMC** styrer hvilken type programmer, der genereres. Værdierne har følgende betydning:

68020	M68020 eller M68030 kode, med Motorola dataudlæg
R3KMI	R3000 kode med MIPS dataudlæg
R3KMO	R3000 kode med Motorola dataudlæg.

cc kræver at TARGETMC-environmentet har en af disse værdier.

2.6 Ny grænseflade til SWD-pakken

Den heterogene SWD-pakke (SWD3.0), afviger fra de foregående, på en række områder, som kan få betydning for andre pakker.

- */bin/ar* og */bin/ld* distribueres fremover *kun* på SWD-båndet; dvs de udgår fra BAS-UTIL-båndet (version 3.0).

Dette betyder at pakker, som benytter *ld* eller *ar* (uden at forudsætte SWD-pakke), fremover selv må levere disse programmer (men selvfølgelig *ikke* lægge dem i */bin!*). Dette betyder også at BAS-UTIL bånd før 3.0 *ikke* må lægges ind efter SWD3.0, da *ld* og *ar* derved overskrives.

- Placeringen af biblioteker (eks: *libc.a*) er ændret, den nye placering er:

```

/lib<TARGETMC>/          og
/usr/lib<TARGETMC>/

```

Hvor <TARGETMC> erstattes af den aktuelle værdi af TARGETMC-environmentet. De lovlige værdier af TARGETMC er: **68000**, **68020**, **R3KMO** og **R3KMI**. Filer svarende til TARGETMC=68000 er ikke længere en del af distributionen for SWD.

Før	Nu
<i>/lib/lib*.a</i>	<i>/lib68000/lib*.c</i>
<i>/usr/lib/lib*.a</i>	<i>/usr/lib68000/lib*.a</i>
<i>/lib20/lib*.a</i>	<i>/lib68020/lib*.a</i>
<i>/usr/lib20/lib*.a</i>	<i>/usr/lib68020/lib*.a</i>
-	<i>/libR3KMO/lib*.a</i>
-	<i>/usr/libR3KMO/lib*.a</i>
-	<i>/libR3KMI/lib*.a</i>
-	<i>/usr/libR3KMI/lib*.a</i>

Med andre ord: biblioteker liggende under */lib* og */usr/lib* bliver *ikke* fundet af linkerens.

Og en bøn på falderebet: lad være at distribuere includefiler i */usr/include*, læg dem i et underkatalog (eks: */usr/include/X11*).

hf [BP]

3. Kodegenbrug i C++

Hvor normal C-kode er 20% genbrug og 80% ny kode, påstår C++ disciple i AT&T at disse procenter kan vendes til 80-20 ved at udvikle i C++. Potentielt kan det betyde kolossale besparelser på udviklingssiden og bedre programmelkvalitet, det sidste p.g.a. genbrug af aftenstet og umodificeret kode.

Denne artikel ser lidt på, hvordan man på et par områder forsøger at opnå genbrug af kode i C++.

3.1 Klasser

I C++ er det muligt at definere nye typer som f.eks. en Pascal-agtig String type. Nye typer defineres v.h.a. klasser, der kan opfattes som en slags indkapslede **struct's** med indbyggede funktioner. Idealet er her, at en sådan klasse skal fungere som en black box, en »dime« man kan bruge i mange forskellige programmer, uden at det kræver noget specielt af programomgivelserne. Et program kan så samles af en række eksisterende, præfabrikerede elementer.

En ny type, som f.eks. `String`, udvider selve sproget og indgår på lige fod med indbyggede typer som `int`, `double` osv. Det eneste man behøver er på normal vis at inkludere "`String.h`" i sit program:

```
/* Programmet demonstrerer
   anvendelse af en String Class */

#include <iostream.h>
#include "String.h"

main()
{
  long tal = 17899;
  String a = "Dette er en streng";
  String b = tal;
  String c;

  c = a;

  if ( c == b )
    cout << c << " er lig med " << b << "0 ; // cout er en print rutine
  else
    if ( c < b )
      cout << c << " er mindre end " << b << "0 ;
    else
      cout << c << " er mere end " << b << "0 ;

  int ital = 10;
  String d = ital;

  c=d;
  if ( d != c )
    cout << d << " er mindre end " << c << "0;
}
```

I `String` klassen er operatorer som `=`, `==`, `>`, `<`, `!=` omdefinerede så de giver mening for en `String`, ligesom output-rutinen, `cout`, har »lært« hvordan den skal skrive en `String` ud. Der er lavet konverteringsrutiner så tal, integers og longs i dette eksempel, kan tilskrives direkte til en `String`. Sammenligningen af to strenge der indeholder cifre kan være alfabetisk, numerisk eller begge dele afhængigt af hvordan `String`-klassen er implementeret.

En anden type vi kunne definere i C++, kunne f.eks. være en Adresse, til brug for programmører i forskellige applikations-miljøer. Hvis vi skelner mellem to typer af adresser, en indenlansk og en udenlansk, ville man i C++ definere tre klasser for en sådan beskrivelse af adresse – først en overordnet type:

```
class Adresse {
protected:
    String          gadeNavn;
    String          lokalTelefonNummer;
    String          byNavn;
public:
    virtual void printOut();
                                // erklæring af print rutine
                                // der er kodet andetsteds
}
```

Klassen indeholder nogen for brugeren skjulte felter, protected og en offentlig printout-rutine.

En hjemlig adresse skal desuden indeholde et postnummer. vi skriver ikke en helt ny klasse, men afleder den fra den eksisterende Adresse:

```
class DanskAdresse : public Adresse {
protected:
    unsigned short postNummer;
public:
    virtual void printOut() { Adresse::printOut();
                            cout << postNummer << "0; }
}
```

DanskAdresse indeholder alle felter fra Adresse plus sit eget postNummer. Den omdefinerer printOut-rutinen til, udover at kalde Adresse-klassens udskrivnings-rutine, desuden at skrive postNummer ud. Det lille ord, virtual, foran Adresse::printOut rutinen markerer, at afledte klasser kan omdefinere denne rutine til eget brug.

En udenlansk Adresse kunne defineres som:

```
class UdenlanskAdresse : public Adresse {
protected:
    String          provins;
    String          nationalPostKode;
    String          land;
    String          internationalPostKode;
public:
    virtual void printOut() { Adresse::printOut();
                            // Klasse specifik kode her }
}
```

Alle felter nedarves fra Adresse, printOut omdefineres og evt. andre funktioner nedarves direkte. En anden vigtig egenskab C++ illustreres her; man modificerer ikke eksisterende kode, man programmerer ved at tilføje ny kode. På den måde undgår man at kaste grus i maskineriet for eksisterende programmer.

Disse Adresse-klasser, læs »typer«, lægges i archives og erklæres i headerfiler, og fungerer som almindeligt tilgængelige sprogudvidelser når de inkluderes.

Hvis en af disse tre klasser ikke tilfredsstillen en programmørs behov, kan han/hun simpelthen aflede sin egen adrestype fra en af de eksisterende.

3.2 Container Classes

Almindeligt anvendte strukturer som arrays, enkelt eller dobbelt-linkede lister, stacks, heaps, hash funktioner m.m. findes allerede som biblioteker til C++. Hertil kommer mere eksotiske ting som dynamiske arrays, pseudo-indexes og Queues m.m. De kan defineres som såkaldte Templates, også kaldt parametricerede typer, der tillader brugeren at anvende dem sammen med forskellige klasser – f.eks. kan en liste defineres som:

```
template <class PT>
class List {
public:
    : // Forskellige funktioner
    : // bl.a. initialisering
    PT append( List<PT>* );
private:
    PT *val;
    List<PT> *next;
}
```

Hvis man erklærer variable af typen:

```
List<Adresse> alist, *palist;
```

bliver `PT` i den ovenstående definition erstattet med den aktuelle type, `Adresse`. `val` bliver en pointer til `Adresse` og `next` er en pointer til næste element i Adresselisten ligesom `append` funktionen kræver en pointer til Adresselisten og returnerer en pointer til `Adresse`.

```
List<int> ilist;
```

erklærer en tilsvarende liste af integers.

Når vi i et programmet skal holde styr på forskellige adressetyper, som de der er beskrevet ovenfor, sker det ved at udnytte den egenskab ved klasser, at de har et indbygget interface, dvs. at de f.eks. selv kender deres `printOut` rutine. Det gør det muligt at opbygge *heterogene* lister af objekter, så længe de har et ensartet interface. Ovenstående Adresseliste kan altså indeholde objekter fra alle typer afledt fra eller af typen `Adresse` og det vil være muligt at skrive kode som:

```
for( Adresse* pa = firstListItem; pa != endOfList; pa = pa->next)
    pa->printOut();
```

C++ vil for hvert objekt vælge den korrekte `printOut` rutine, for netop den type objektet har.

Det er naturligvis muligt at skrive noget lignende i C, men det vil sandsynligvis involvere typefelter og switches på disse, switches der sandsynligvis skal rettes til hver gang nye typer skal puttes ind i listen. I C++, hvis `append`-funktionen for `Adresse` er anderledes end den eksisterende standardfunktion, tilføjes den simpelthen som en type-specifik funktion:

```
Adresse* List<Adresse>::append( List<Adresse>* pt)
{ /* ... */ }
```

C++ tilbyder faciliteter der hjælper en, når man ønsker at kode moduler som kan genbruges. Genbrugelig kode er ikke let at skrive, selv ikke i C++, og selve denne objektorienterede tilgang til programmering synes at tvinge én til flere overvejelser og bedre planlægning, før man koder derudaf – og det er jo en god ting.

cfj [BP]

4. Indtryk fra Look91 Konferencen

Fra 2-2 til 4-2-91 deltog cfj (Claus), es (Eggert), hf (Helle) og cbp (Christian) i Look91 Konferencen om Object Orienteret Programmering og Metode, der afholdtes på Marienlyst i Helsingør.

4.1 Hvad tilbyder Objekt Orienteret Programmering

I OOP (objekt orienteret programmering) fungerer hvert enkelt objekt som en *black box* der tilbyder bestemte faciliteter. Data og intern funktionalitet er skjult for brugeren (udvikleren), der derimod har et interface til objektet bestående af funktionskald. Dette objekt kan så bruges som byggeklods i forskellige systemer.

I OOP koder man ikke ved at ændre objekter, men ved at *tilføje* ny funktionalitet til eksisterende objekter og på den måde skabe nye objekter. Det betyder, at et en gang kodet og aftestet objekt ikke kodes om, selv når det indgår i nye sammenhænge.

Gevinsterne er:

- Genbrug af kode — hvor traditionel C-programmering kun anvender omkring 20% genbrug, kan denne procentdel ifølge optimisterne hæves til 80% i et C++ miljø, når først de firma-specifikke objektbiblioteker er udviklet.
- Reducerede udviklingsomkostninger og mindre risiko. Nye systemer kan opbygges af og ud fra eksisterende og aftestede objekter, ligesom tilføjelser til objekter ikke introducerer fejl i gamle systemer.
- Lavere vedligeholdelsesomkostninger.
- Kortere tid til markedet. Et produkts kommercielle succes kan naturligvis afhænge af, om det kommer på markedet på det rette tidspunkt — om det rammer, hvad amerikanerne kalder *The Marked Window*.
- Bedre debugmuligheder, da objekter typisk kun kan modificeres igennem bestemte af sine interfacefunktioner.
- Indkapslingen af data og intern funktionalitet i objektet, tillader rettelser i objektet, så længe det basale interface til

objektet ikke ændres. Programmerer der benytter objektet skal da måske genlinkes, men ikke omkodes.

For de folk der taler om et paradigme skift i forbindelse med OOP, er idealet VVS-manden, der samler et centralvarmesystem ud fra præfabrikerede elementer. Man taler om en *Software Industrial Revolution*, hvor det at skrive software grundlæggende benytter sig af masseproduktions-teknikker, og kan opfattes som en slags *assembly*-aktivitet.

4.2 OO-Sprog

De objekt orienterede sprog deler sig i to grupper:

1. **Pure Languages** som Smalltalk, Eiffel — disse sprog har en trofast skare af brugere i det akademiske miljø og enkelte kommercielle firmaer der bruger dem som udviklingssprog. Sprogene har egenskaber som:
 - udtryksfuldhed, dvs. gør det nemmere for brugeren at beskrive sit system.
 - De tillader en **Late Binding**, at referencer først løses på runtime, på bekostning af typecheck. *Late Binding* betyder at systemer som Smalltalk i en hvis udstrækning arbejder som en fortolker. Prisen er dårligere performance — op til 400%.
 - De leveres med en værktøjskasse af standardobjekter.
2. **Urene sprog** som C++ der er kendetegnet ved:
 - god integration med det omgivende UNIX miljø. C++ kodede systemer kan integreres med C-kode eller andre programmeringssprog. På den måde kan man sikre investeringen i eksisterende systemer.
 - Meget stærk typecheck på compilerings-tidspunktet. Dette punkt alene kan sikre en langt lavere fejlfrekvens i programmerne.
 - C++ er præget af en tankegang der siger, at afviklingshastigheden skal være som i C. OO-faciliteter

der ikke kunne leve op til denne forudsætning er simpelthen ikke blevet implementeret. Resultatet er et sprog der kun er ca. 10% langsommere end C.

Det er vores klare indtryk, at hvor den gamle garde af OO-tilhængere ikke er særlig begejstrede for C++, så vil næsten al seriøs udvikling inden for OO i de kommende år ske i C++. Alle større EDB-virksomheder satser den vej i deres OO-strategi, ligesom udviklingsmiljøet nu er blevet stabilt.

4.3 Udviklingen fremover

Afgørende for C++'s popularitet i 90'erne vil være tilgængeligheden af objektbiblioteker. I løbet af det sidste års tid er der dukket en del pakker op, så udviklingen er ved at tage fart. Man kan skelne mellem 3 typer af objektbiblioteker:

- Servicepakker — grafiske biblioteker og window-biblioteker findes allerede på markedet. Et for DDE aktuelt eksempel er, OI-biblioteket fra AT&T der tillader udvikleren at kode imod Open Look og Motif igennem ét C++ interface. 30% procent af markedet skønnes at blive denne slags pakker.
- Industripakker — bank, produktion og forsikring pakker er så småt begyndt at dukke op. For det meste er der tale om *Application Frameworks*, bestående af relevante objekter, som så kan udvides og tilpasses til de konkrete behov. Pointen er her, at man ikke køber færdige applikationer, men snarere byggeklodser til applikationer. 30% af markedet skønnes at blive applikationspakker.
- Organisationspakker — DDE vil typisk kunne drage nytte af standardiserede firma-specifikke funktioner og objekter lagt i biblioteker. På den måde er det muligt at sikre en højere grad af software-genbrug og kvalitet i produkterne. 30% af udviklingsindsatsen skønnes at ligge på denne type af programmel.
- De sidste 10% dækker over specielle anvendelser.

4.4 Objekt Orienteret Metode

På konferencen skortede det ikke på advarsler: *Denne teknologi vil ikke løse alle*

deres problemer og Indført på en forkert måde, uden tilstrækkelig uddannelse og de rigtige organisatoriske strukturer, vil den medføre katastrofer.

En stor del af konferencen handlede om metode. En vigtig indsigt i forbindelse med OOP er, at hovedproblemet ikke er syntaksen i det nye sprog, men metoden — det at vende sig til at tænke systemer som hierarkier af objekter. Almindelig strukturel metode, hvor der opdeles i data for sig og procedurer for sig, saboterer nærmest en OO-tankegang, hvor data og funktionalitet netop er koblet så tæt som muligt.

Skal man indføre OOP i organisationen, er det ikke gjort med at lære folk sproget alene.

4.5 Objekt-Databaser

Objekt Databaser har nået vis modenhed og er ved at være kommercielt interessante. I C++ er en ODB typisk en integreret del af sproget. Det at udføre DB-operationer på et objekt er en integreret del af objektets interface, ligesom ODB-objektet har indbygget viden om hvilke relationer det har til andre objekter i databasen.

En grundlæggende ide i ODB's er, at repræsentationen af et objekt skal være den samme i databasen som i programmet. At ændre repræsentationen af en datamængde, f.eks. for at lagre den i en relationsdatabases normaliserede tabeller, er med et citat fra konferencen »som at skille sin bil ad hver aften, når den skal i garage«. For visse typer af applikationer, f.eks. indenfor CAD/CAM og grafik, er ODB begyndt at have i vis udbredelse, netop fordi de i disse sammenhænge kan være op til 4-6 gange hurtigere end en relationsdatabase (tallet stammer fra Thomas Atwood, der leder firmaet Object Design).

Et produkt som ObjectStore, Object Disigns ODB, er et interessant bud på en Object Orienteret database. Den udvikles i samarbejde med AT&T og er fuldstændig integreret med AT&T C++ præprocessor. Af andre firmaer der er ønsker at tilbyde den i deres OO-miljø er Sun, Hp/Apollo, DEC, IBM, NCR, Sony, Olivetti, Microsoft og Borland.

cfj [BP]

5. CSCW – Computer Supported Cooperative Work

Rapport fra den 3. internationale konference i CSCW 7-10 oktober 1990 i Los Angeles, California.

CSCW er betegnelsen for det edb-understøttede samarbejde. Denne betegnelse dækker over en relativ ny orientering inden for edb-verdenen, hvor fokus har ændret sig fra en person – en opgave – en computer til edb-understøttelse af flere personer, der gennem et gensidigt samarbejde løser en eller flere opgaver.

Hovedindfaldsvinklen for CSCW er, at opgaver hovedsageligvis løses ved, at folk arbejder sammen i grupper. Dette kan typisk være mindre grupper – evt. sammensat i en større organisation (som DDE's organisering i afdelinger og grupper), eller løst strukturerede grupper, dvs. grupper, der går på tværs i organisationen og ikke kan aflæses i et organisationsdiagram (som fx gruppen af applikationsudviklere i DDE).

Forskningen inden for CSCW er centreret omkring den tese, at edb-teknologien kan blive et effektivt middel til at forandre den måde, som folk samarbejder på. Forskningen forsøger at besvare spørgsmål omkring, hvordan store og små gruppers samarbejde kan understøttes af edb-teknologien:

- Hvordan skal arbejdssituationer analyseres for, at samarbejdet og problemer og »flaskehalse« deri blotlægges?
- Hvilken type edb-teknologi og hvilke typer software dertil skal udvikles, for at understøtte og effektivisere dette samarbejde?
- Hvordan skal man planlægge at arbejde sammen for at få fordel af edb-teknologien og den tilknyttede software?

Forskere og udviklere fra flere forskellige områder begyndte at interessere sig for sådanne spørgsmål i starten af 1980'erne. Den første workshop om CSCW blev holdt på MIT i august 1984 og den første egentlige internationale konference blev afholdt i Austin i december 1986. Denne konference havde 300 deltagere. 2. konference blev afholdt i Portland i september 1988. På den 3. konference i Los Angeles i oktober 1990

var der registreret over 560 deltagere fra en mængde forskellige nationer. 4. konference er planlagt i 1992 i Toronto. Der er sideløbende blevet afholdt en »European Conference on CSCW« og den 2. E-CSCW afholdes i september 1991 i Amsterdam.

Forskningsprogrammet er interdisciplinært i sin natur, idet der fokuseres på CSCW ud fra både tekniske, sociologiske, organisatoriske, erkendelsesmæssige og arbejdsmæssige vinkler. Deltagerne i konferencerne spænder over forskellige forskningstraditioner inden for bl.a. computer science, management og organisation, cognitive science, AI, sociologi, antropologi og psykologi.

Det er ikke muligt at give en udtømmende beskrivelse – eller afgrænsning af udviklingen inden for CSCW i dag, på baggrund af konferencen i Los Angeles: CSCW kan kortfattet præciseres som et *perspektiv*, som systemudviklere og brugere kan anvende ved analyse, design, implementering og brug af edb-baserede systemer. Et perspektiv der – som nævnt – fokuserer på det edb-understøttede samarbejde.

I det følgende vil jeg belyse dette perspektiv ved at beskrive

1. nogle typiske »rendyrkede« CSCW-produkter,
2. anvendelsen af dem i »the real life« samt
3. nogle teoretiske og metodiske overvejelser ved udvikling og brug af CSCW-produkter.

Afslutningsvis vil jeg knytte nogle få kommentarer til

4. hvad DDE så kan lære af det.

5.1 Produkterne

Der blev på konferencen gennemgået en lang række helt eller delvist færdigudviklede CSCW-produkter – eller groupware, som de også benævnes.

Nogle af disse produkter kan kategoriseres inden for områderne elektronisk post, konferencesystemer, beslutningsstøttesystemer og fælleseditorer.

Det »ældste« CSCW-produkt, som stadig er nøgleproduktet, er *elektronisk post*.

Produkter, der — som elektronisk post — kan understøtte grupper, der samarbejder adskilt i tid og rum hører til de mest udviklede CSCW-produkter. Eksempler på disse er:

- *Strudel* — An Extensible Electronic Conversation Toolkit. Strudel er et UNIX/X-window/Motif-baseret toolkit til strukturering af konversationer (kommunikation via elektronisk post) i grupper. Fx kan Strudel afbilde et konversationsforløb grafisk. Elektronisk post kan med Strudel tilføjes definitioner for:

- Konversationer: Hvilken konversation (ID) hører dette elektroniske brev til.
- Konversationstræk: Hvilken type konversation er dette, fx en forespørgsel, et svar på en forespørgsel m.m.
- Aktioner: Fx »Sæt tid af til et møde«, »Noter på huskeseddel«, »Opdater systemdokumentation efter fejlretelse«, »Fakturer i økonomisystem« osv.
- Memo: Beskrivelse af en aktion, som en bruger af Strudel vil udføre (med tid, sted, prioritet m.m.)

En af hovedideerne i Strudel er, at brugerne af systemet selv definerer deres konversationer, konversationstræk, aktioner og memo'er. Strudel er dermed et værktøj, der understøtter en selvdefineret struktureret konversation i en gruppe.

- *The Coordinator*, der er et kommercielt produkt, som bruges af mere end 100.000 brugere, er et system til elektronisk post, opbygget ud fra en sproghandlingsteori formuleret af J. R. Searle. Denne teori hævder, at al kommunikation (speech act — sproghandlinger) kan operere inden for nogle få »kversationsblokke« (sproghandlingskategorier) — fx:

- Request/promise
- Offer/acceptance
- Report/acknowledgement

Ved brugen af *The Coordinator* er det intentionen, at afsenderen overvejer, hvilken type sproghandling vedkommende sender af sted. Systemet understøtter, at al elektronisk post

afsendes som tilhørende en af de nævnte sproghandlingskategorier (+ en »free form«-kategori), og systemet linker automatisk al til- og fra-post til deres respektive tilhørende konversationer.

Konferencesystemerne hører til den mere sofistikerede produkttype, hvor teknologien ofte omfatter både audio- og visuelle medier. Nogle af disse systemer forsøger at understøtte mødesituationer, hvor deltagerne befinder sig lang fra hinanden. Eksempler på dette er:

- *The VideoWindow teleconferencing system*. Dette system forsøger at understøtte informel kommunikation ved telekonferencer. Systemet simulerer, at 2 rum, der faktisk befinder sig langt fra hinanden, er ét rum, opdelt med en glasvæg i midten. Til dette benyttes videosystemer med storskærme på næsten 1 meter gange 2,5 meter. Dette skal give deltagerne den fornemmelse, at de står over for hinanden og konfererer.
- *MERMAID: A Distributed Multiparty Desktop Conferencing System*. Dette system er baseret på arbejdsstationer. Med en sådan arbejdsstation foran sig kan deltagerne fjernt fra hinanden kommunikere sammen gennem videobilleder, stemmer, tekst, grafik, »still-billeder« og figurer tegnet i hånden. Hver arbejdsstation indeholder videokamera, mikrofon, højtalere, scanner, mus og elektronisk pen samt en grafisk vinduesbaseret skærm, der kan vise billeder i bevægelse, still-billeder samt alm. grafik og tekst præsenteret i enten »fælles« eller i »private« vinduer på skærmen.

Beslutningsstøttesystemer er produkter, der understøtter en visuel repræsentation og håndtering af viden ifm. beslutningsprocesser. Hovedideen i disse systemer er dels at registrere denne viden til senere evaluering og genbrug og dels at gennemsigtiggøre beslutningsprocessen for deltagerne heri.

- *SIBYL* er en videreudvikling af et system kaldet gIBIS (Graphical Issue Based Information System). SIBYL understøtter repræsentation og håndtering af beslutningsprocessens alternativer, mål og de argumenter, der evaluerer alternativerne i forhold til de opstillede mål. SIBYL består af sproget DRL (Decision Representation

Language), et sæt funktioner, der arbejder på data repræsenteret i DRL samt en grafisk brugergrænseflade. Via DRL kan man:

- Definere problemer.
- Opstille alternative løsninger.
- Beskrive mål og delmål.
- Give påstande, der kan understøtte, tilbagevise eller forudsætte mål eller andre påstande.
- Stille spørgsmål til påstande som – evt. – kan påvirke en påstand, hvis dennes vægt afhænger af svaret på spørgsmålet.

SIBYL genererer en vidensbase, der kan visualiseres via en »beslutningsgraf«, der viser beslutningsprocessen mål, delmål, alternativer, påstande, spørgsmål og relationerne mellem disse.

Mål og alternativer for et problem opstilles i SIBYL i en beslutningsmatrix, hvor alternativernes relation til målene kan evalueres.

Brugernes (beslutningsdeltagernes) argumentation for og imod opstillede påstande visualiseres ved dele af den genererede beslutningsgraf.

SIBYL's funktioner understøtter:

- Referenceintegriteten i vidensbasens relationer.
- Viewpoints, hvorigennem fx effekten af forskellige vægte til et givet mål kan aflæses i beslutningsgrafen.
- Genbrug af tidligere beslutningsprocesser: Fx hvis et nyt problem har mål til fælles med et tidligere registreret problem, kan relationerne til disse mål trækkes ind i det nye problems beslutningsgraf.

Fælleseditorer er værktøjer til understøttelse af flere personers samtidige skrivning/kommentering af tekster, fx rapporter, artikler eller programkode. En række (fortrinsvis Macintosh-baserede) fælleseditorer blev demonstreret (muligvis var dette produkt meget vel repræsenteret, idet fælles skrivning af fx artikler er en stor aktivitet blandt forskere?).

- *ShrEdit*, Shared Editor er et resultat fra et forskningsprojekt, der fokuserer på folk, der arbejder i de tidlige stadier af et systemdesign. (Produktet kan dog lige såvel understøtte fx tilbudskrivning, hvor flere personer skriver simultant på samme dokument). ShrEdit giver flere personer mulighed for samtidigt at arbejde med en eller flere fælles tekst-filer. Med ShrEdit kan man have tekster fra en fælles fil fremme i et vindue på skærmen: Man kan vælge det afsnit ud af teksten, som man vil editere. Hvis ingen andre arbejder med dette afsnit får man lov og »låser« samtidig afsnittet, så andre ikke kan editere det. At en anden har låst et afsnit i teksten kan ses ved, at denne tekst er mørkere end den »frie« tekst. Der kan derudover åbnes »public«-vinduer til kommentarer, som alle kan se og »private«-vinduer til egne private noter.

- *ICICLE*, Intelligent Code Inspection Environment in a C Language Environment er et (X-Window-baseret) system til understøttelse af kode-inspektionsmødet, der er en aktivitet, som kan finde sted mellem implementation og afstætning af (større) systemer udviklet i C. (Flere organisationer – bl.a. IBM, AT&T, og Bellcore – har de seneste år eksperimenteret med formel kode-inspektion. Kode-inspektionens detektering af fejl inden afstætningen har givet ressourcebesparelser, der fx har medført, at Bellcore's Software Technology and Systems Area har sat sig som mål at udføre kode-inspektion på 100% af deres nyudviklede software.)

ICICLE-systemet understøtter både forberedelsen af kommentarer til – og selve kode-inspektionsmødet om den C-kode, der skal inspiceres. ICICLE har til dette funktioner til detektion af rutinefejl (bl.a. baseret på standardanalyseprogrammer som lint), krydsreferencecheck og vinduesstyring af den kode, som mødet er genstand for.

Deltagerne i mødet kan kommunikere alene gennem deres arbejdsplads med ICICLE: En person styrer et bestemt vindue fælles for alle skærme, der viser den del af koden, som der er fokus på. Alle deltagere kan åbne kommentar-vinduer, med kommentarer til en del af koden og

aktivere knapper i kommentar-vinduet, for at angive, om de accepterer eller afviser kommentaren. Godkendte kommentarer lagres som et resultat af mødet.

Et gennemgående træk ved alle de fremviste produkter på konferencen var, at *samtliche* produkter var baseret på grafiske brugergrænseflader. De fleste baseret på Macintosh, nogle på X-Window og enkelte på MS-Window.

Fælles for hovedparten af de ovenfor beskrevne produkter er, at de er standardprodukter, der er *specialiserede* til at understøtte funktioner i samarbejde som elektronisk post, tele-konferencer, beslutningsprocesser, fællesskrivning osv. At fokusere på disse produkter alene vil efter min opfattelse give et forvrænget indtryk af CSCW — idet CSCW set som et *perspektiv* er mere end disse »rendyrkede« CSCW-produkter: CSCW er i lige så høj grad elementer og dele af mere traditionelle administrative edb-produkter samt analysen, designet, implementeringen og brugen af disse.

5.2 Anvendelsen af CSCW-produkter

Anvendelsen af CSCW-produkter i forskellige organisationer er blevet analyseret i en større undersøgelse foretaget af Christine V. Bullen fra MIT Sloan School og John L. Bennett fra IBM Research Division.

Undersøgelsen, der er foretaget over en årrække, omfatter interviews med 223 brugere af CSCW-produkter fra 25 forskellige firmaer i størrelser varierende fra under 1 mill. \$ til 30 milliarder \$ i omsætning. Produkterne, der indgik i undersøgelsen var The Coordinator Version 1, PROFS, ForComment, All-in-1, Higgins, Metaphor og In-House System 1 og 2.

Undersøgelsen gik ud på at betragte og analysere forskellige brugere i forskellige typer organisationers brug af — og subjektive mening om de CSCW-produkter, som de havde til deres rådighed. En sådan type undersøgelses resultater kan efter min mening generaliseres til andre produkttyper end netop CSCW-produkter.

Undersøgelsens hovedkonklusioner var følgende:

Fra et designperspektiv:

- **Elektronisk post-kommunikationen er det primære værktøj.** Det værktøj, som var mest benyttet, var elektronisk post. Hvis man kan antage, at brugerne kun benytter de dele af systemerne, der er væsentlige for deres arbejde, er konklusionen, at edb-understøttelse af elektronisk post er mest nødvendigt.

- **Linkning af til- og fra-post er en nøgelfunktion i elektronisk kommunikation.** En nøgelfunktion i brugen af elektronisk kommunikation er evnen til at linke meddelelser til et bestemt emne eller til at associere meddelelser med en eller anden form for distributionsliste. Det er vigtigt, at man kan slå op eet sted, for fx at se al diskussion om et bestemt projekt. Den automatiske linkning af meddelelser til den tilhørende »konversation«, som produktet The Coordinator tilbyder, var anset for et af dette produkts væsentligste egenskaber. Egentlig er der overraskende, at en sådan funktion er forholdvis primitiv i langt de fleste edb-produkter til elektronisk kommunikation: Det har altid været naturligt for folk at forsøge at organisere de — ofte meget store — mængder information, som skal håndteres.

- **Hvilken funktionalitet der er inkluderet og hvordan den benyttes er vigtige faktorer.** Et af de bedste eksempler på en en meget efterspurgt funktionalitet, der ikke kan benyttes effektivt er den elektroniske kalender. Den elektroniske kalender har en potentiel understøttende funktion ifm. gruppeaktiviteter, men de nuværende systemer kan ikke erstatte den traditionelle papirkalender:

- Papirkalenderen har langt flere funktioner end den elektroniske kalenders mødetidsregistreringer: Fx avancerede emneopdelinger, telefonlister, gule klæbe-sedler med notater, noter og breve sat fast med klips m.m.

- Elektroniske kalendere er ikke til at have med sig overalt.

- **Isolerede værktøjer sænker effektiviteten af groupware-systemer.** Værktøjerne kan være isolerede, eller mangle integration på 2 måder:

- Ifm. hvordan de benyttes, styres og kontrolleres af brugeren. Forskellige måder at udforme brugerinterface på og besværlige procedurer for at skifte fra et værktøj til et andet er eksempler herpå. Et standardiseret brugerinterface og enkle rutiner til skift mellem diverse værktøjer, som fx Macintosh-produkterne konsekvent benyttes, er væsentlige effektivitetsfaktorer.
- Ifm. transport af data fra et værktøj til et andet. Selv enkle rutiner til at skrive data midlertidigt ud fra et værktøj, for derefter at læse samme data ind i det næste opleves som en unødvendig besværlighed. Den højeste grad af integration er her et krav, og et ideal ville være, fx at tekster fra de forskellige, men personligt foretrukne tekstbehandlingsanlæg umiddelbart kunne flyttes mellem hinanden.

Der blev på konferencen ofte stillet det dobbelte idealkrav, at folk skulle kunne benytte deres foretrukne værktøjer (da man er uvillig til at skifte disse ud med andre, og fordi det også er en stor omkostning at skulle blive fortrolig med nye værktøjer) samtidig med, at værktøjerne som helhed skulle være »seamless«, dvs. skulle tilbyde en gnidningsfri overgang fra produkt til produkt, både mht. brugerinterface og datatransport.

Fra et organisatorisk perspektiv:

- **Den største værdi fra edb-værktøjer findes, hvor de parallelliserer tilsvarende ikke-elektroniske aktiviteter.** Grunden til, at elektronisk post er meget benyttet og populært skal ses i, at elektronisk post er helt analog til en manuel postfordeling, blot mere effektiv, idet det er hurtigere, muligt at spore tilbage, tid- og sted-uafhængig, og muligt at nå næsten overalt, selv hjemme eller på rejser. Det er derfor nemt at indse fordelene i at skulle lære at benytte elektronisk kommunikation. Funktioner, der er væsentligt forskellige fra tilsvarende ikke-elektroniske aktiviteter (fx den elektroniske kalender) eller funktioner, som kun sjældent benyttes, anses ofte for at være mere ressourcekrævende at udføre med edb-værktøjer end uden.

- **Der skal være balance mellem fordelene ved at benytte edb-værktøjer og ressourcerne, der skal investeres for at benytte dem.** Fx oplever de fleste brugen af elektroniske kalendere som redundant og trivielt indtastningsarbejde, også selv om det erkendes, at ledelsen eller gruppen (de andre) havde fordel af det. Den elektroniske kommunikation opleves derimod som ren »indtægt«: Med elektronisk post »får man sine meddelelser ud«, »får kastet bolden over hos en anden«, uddeleret arbejde til gruppens medlemmer osv.

Selv om man godt kan se en evt. fordel i brugen af et edb-værktøj er folk generelt uvillige til at ændre adfærd og arbejdsgang og tilpasse den edb-værktøjet. Fordelene ved at benytte edb-værktøjet skal klart synes større end ressourcerne, der skal investeres i at lære og regelmæssigt anvende dette. Klare motivationsfaktorer, der kan forrykke denne balance er fx ledelsesdirektiver, enighed i gruppen og uddannelse i brugen af værktøjerne.

- **Introduktionen til nye edb-værktøjer er en vigtig faktor.** Undersøgelsen viste, at i situationer, hvor ledelsen forventede væsentlig produktivetsforøgelse ved (næsten) blot at installere et CSCW-produkt, var oplevelsen hos brugerne, at de ikke fik nogen signifikant ændring i deres tilgang til deres arbejde, og nogle mente, at den nye teknologi blot gav dem mere arbejde og sænkede deres produktivitet. Specielt introduktionen af nye edb-værktøjer er meget kritisk:

- De forventninger, som brugere af edb-værktøjerne får ved introduktionen til disse, påvirker brugen af dem lang tid fremover.

- Ved uddannelsen eller træningen i brugen af edb-værktøjer er det vigtigt, at de underliggende ideer og tænkte funktionaliteter ved værktøjet formidles. Ofte minimeres denne introduktion til værktøjet til en mekanisk gennemgang af, »hvordan der trykkes på knapperne«.

- Den mest benyttede måde, at erfare nye muligheder med et edb-værktøj, efter, at man har fået det introduceret

og benytter de mere basale funktioner, er gennem en kollega eller underordnet person, der har andre/flere erfaringer med værktøjet end én selv. Kun sjældent slås der op i manualerne for at få hjælp, og de fleste kan ikke finde den information, de søger i evt. on-line hjælp.

En opmærksomhed på disse kritiske punkter har kun sjældent været været inddraget ved organisationens introduceren til nye edb-værktøjer.

Undersøgelsen konkluderer, at nogle af de ovenstående betragtninger kan skyldes, at CSCW-applikationer er en relativt ny teknologi. Samtidig fastslås, at nogle af CSCW-produkterne – som fx The Coordinator – hvis ellers introduceret og benyttet efter designernes anbefalinger, vil være effektive værktøjer til at reducere og simplificere mængden af informationer med det resultat, at brugerne ville kunne koncentrere sig mere om indholdet og meningen med interaktionen end om at håndtere og administrere mængden af informationerne.

De fleste af disse observationer mener jeg ikke er ikke specifikke for CSCW-produkter, derimod gælder de softwareprodukter generelt.

5.3 Teoretiske og metodiske overvejelser

Teori og metodeudviklingen halter (naturligvis?) bagefter produkterne, men det var en entydig konklusion af konferencen, at en succesfuld videre- og nyudvikling af CSCW-produkter har stærkt brug for teoretiske og metodiske afklaringer.

CSCW-produkterne knytter sig op ad flere forskellige forskningstraditioners perspektiver, der hver især beskriver dele af CSCW-konceptet uden dog, at hele dette koncept er sammenhængende beskrevet og forstået. CSCW-produkterne kan kategoriseres inden for mindst 4 retninger:

Produkt eks.	Forfatter (designer)	Forskningstradition	Koncept
The Coordinator	Winograd et al.	Lingvistik	Sproghandlinger (Speech acts)
gIBIS	Conklin & Begeman	Rhetorik	Argumentationsstrukturer
Together	Holt et al.	Computer Science	Procesrepræsentation
Information Lens	Malone et al.	Organisations-terori og AI	Adhocrater

Der præsenteredes kun et par indlæg, der fokuserede på begrebsafklaring, bl.a. fra Thomas W. Malone, MIT. Malone præsenterede en indledende afklaring af, hvad man skal forstå ved »Cooperative work«. Han fokuserede på koordineringsaspektet idet han fremførte, at gennem en bedre forståelse af, hvad koordination er, opnås muligheder for at udvikle brugbare edb-værktøjer til understøttelse af samarbejde. Han definerede koordination som:

”The act of managing interdependencies between activities performed to achieve a goal”

og fastslog samtidig, at:

”If there is no interdependence, there is nothing to coordinate”.

Malone gennemgik en indledende ramme, inden for hvilken flere forskningstraditioner kan bidrage til en yderligere teoriudvikling omkring koordination. Denne ramme – som jeg ikke vil beskrive nærmere her – omfatter bl.a. en kategorisering af »interdependencies« og en skitsering af de underliggende processer for koordination.

Hvad angår metoder til udvikling af CSCW-produkter var der kun få indlæg.

Claudio Ciborra fra Universita de Trento (Italien) var bidragsyder til et af dem, og han argumenterede imod brugen af konventionelle strukturerede metoder, som fx Yourdon's metode med data flow-diagrammer, til design af CSCW-produkter, idet:

En forbedring af samarbejdet i et team afhænger mere af sammenhængen af team-medlemmerne (fx højere tillid, større motivation, bedre flexibilitet m.m.) end af

tilvejebringelse af flere informationer og data.

- CSCW-produkter skal ikke kun understøtte data, men også tekst, stemmer og billeder som væsentlige dele af samarbejdet mellem mennesker.

Ciborra foreslog at fokusere på det »netværk af transaktioner mellem individuelle og organisatoriske enheder« som opretholdes af kontrakter, hvis natur afhænger af niveauet af tillid mellem de individuelle parter og kompleksiteten af de opgaver, der skal udføres.

Hertil omtalte han »The transaction cost method«, der er en systemudviklingsmetode inspireret af teorier og metoder inden for økonomien. Denne metode identificerer og beskriver transaktioner i en organisation, tager udgangspunkt i de flaskehalse, huller, »breakdowns« og ineffektiviteter, som er karakteristiske for disse transaktioner, og benytter dette som grundlag for design af edb-systemer, der understøtter en forbedring af transaktionerne og/eller en forandring af deres indbyrdes organisering. Metodens perspektiv er således mere *hvordan* processer forløber i en organisation end *hvad* indholdet af disse processer er.

Alt i alt ser der ud til, at den metodeudvikling, der de sidste 10 år har foregået på datalogiske institutter i Skandinavien vil være af stor interesse for CSCW-forskningsprogrammet, og det sidste indlæg på konferencen afsluttede da også med, som en af hovedkonklusionerne, at sige:

«Let's go Scandinavian».

Interessen drejer sig her om metoder til

1. Kvalitative analyser af brugernes arbejde.
2. Effektivisering af samarbejdet mellem brugere og udviklere mhp. at generere design ideer.

Flere indlæg strejfede behovet for nye metoder, men de var generelt ikke særlig konkrete. Argumentationen var dog, at den type edb-produkter, som går dybt ind i en og flere personers arbejde, forudsætter en grundig forståelse af dette arbejde, for ikke at skabe ekstra besvær og problemer.

Som alternativer til eksisterende metoder blev observation og andre antropologisk inspirerede metoder nævnt. Problemet er (blot), hvordan de kan omformes, så de kan indgå i en designproces og (helst) også indgå på en sådan måde, at der kommer brugbare systembeskrivelser ud af det.

5.4 Hvad kan DDE så lære af det

DDE må være interesseret i at skabe nye produkter samt til at videreudvikle eksisterende produkter til at supplere sin løsningsstrategi.

5.4.1 Produktudvikling

At tænke i CSCW-termer har ledt til nye produkter, som der ser ud til at være et marked for. Dette kræver dog eksperimenter i *virkelige* omgivelser. DDE har tidligere kunnet gennemføre sådanne eksperimenter med eksterne samarbejdspartner, fx omkring udviklingen af kontorsystemerne i samarbejde med Århus Tekniske Skole. Nogle nye produkttyper, fx fælleseditorer som beskrevet ovenfor, og måske også projektkoordinerings- og projektstyringsværktøjer, der tager udgangspunkt i, hvad koordinerings- og styringsproblemerne *faktisk* er for projektgrupper og for DDE's ledelse, kunne være produkter, hvor DDE *hos sig selv* har den virkelighed, der skal til, for at eksperimentere med udviklingen sådanne.

5.4.2 Evaluering af egne produkter

Med henblik på feedback til videreudviklingen af DDE's egne produkter kunne CSCW-inspirerede analyser af, hvordan DDE's produkter *faktisk* anvendes af kunderne være en god ide.

5.4.3 CSCW som perspektiv på udvikling af applikationsprogrammer

At betragte arbejdsopgange i et CSCW-perspektiv sætter fokus på andre dele af arbejdet, end man måske er vant til. Dette kan lede til nye designideer.

Ex vil et CSCW-perspektiv på en universitetsadministration bl.a. forsøge at belyse kommunikationen mellem den centrale administration og den decentrale administration på institutter og afdelinger. Dette perspektiv mener jeg vil fokusere mere på *måden*, hvorpå den centrale og decentrale administration fordeler og administrerer kompetencer og arbejdsopgaver. Et mere traditionelt perspektiv vil i højere grad fokusere

på de informationer, som (i første omgang den centrale) administration burde registrere og have adgang til.

5.5 Litteratur

Nedenstående litteratur kan fås til kopiering hos jsi/493:

CSCW'90 Los Angeles: Proceedings of the Conference on Computer-Supported Cooperative Work, October 7-10, 1990 Los Angeles, CA
Sponsored by ACM, SIGCHI & SIGOIS

Ad 1) Produkterne:

- Strudel:

Strudel - An extensible electronic conversation toolkit
A Shepherd, N. Mayer, A. Kuchinsky
Hewlett-Packard Labs
Proceedings of... pp 93-104

- The Coordinator:

Understanding Computers and Cognition
A New Foundation for Design
T. Winograd, F. Flores
Addison-Wesley, New York, 1986

- Desuden omtalt i:

Learning from user experience with groupware
C. W. Bullen, MIT Sloan School
J. L. Bennett, IBM Almaden Research Center
Proceedings of... pp 291-302

- The VideoWindow teleconferencing system: The VideoWindow System in informal communication

R. S. Fish, R. E. Kraut, B. L. Chalfonte
Bellcore
Proceedings of... pp 1-12

- MERMAID:

Distributed multiparty desktop conferencing system: MERMAID
K. Watabe, S. Sakata, K. Maeno, H. Fukuo, T. Ohmori
NEC Corporation
Proceedings of... pp 27-38

- SIBYL: SIBYL: A tool for managing group design
J. Lee
MIT

Proceedings of... pp 79-92

- gIBIS: Report on a development project use of an issue-based information system

K. C. B. Yakemovic,
NCR Corporation
E. J. Conklin,
MCC

Proceedings of... pp 105-118

- ShrEdit: ShrEdit: A shared editor for the Macintosh

Papir fra Cognitive Science and Mascine Intelligence Laboratory
University of Michigan

- Anden litteratur om fælleseditorer:

Concurrent editing: The group's interface

J. S. Olson, G. M. Olson, University of Michigan

L. A. Mack, Andersen Consulting
P. Wellner, Rank Xerox Euro-PARC

Issues in the design of computer support for co-authoring and commenting

C. M. Neuwirth, D. S. Kaufer, R. Chandhok, J. H. Morris
Carnegie Mellon University
Proceedings of... pp 183-196

- ICICLE:

ICICLE: Groupware for code inspection

L. Brothers, V. Sembugamoorthy, M. Muller
Bellcore

Proceedings of... pp 169-182

Ad 2) Anvendelsen af CSCW-produkter

- Learning from user experience with groupware

C. W. Bullen, MIT Sloan School
J. L. Bennett, IBM Almaden Research Center
Proceedings of... pp 291-302

Ad 3) Teoretiske og metodiske overvejelser:

- Some social and economic consequences of groupware for flight crew

I. Benson, University of Cambridge
C. Ciborra, Universita di Trento & Institute

- Theseus
S. Proffitt, British Airways
Proceedings of... pp 119-131
- User support: Illustrating computer use in collaborative work contexts
R. Hellman
University of Turku
Proceedings of... pp 255-268
 - Interface
J. Grudin
Aarhus University & MCC
Proceedings of... pp 269-278
 - What is coordination theory and how can it help design cooperative work systems?
T. W. Malone, K. Crowston
MIT
Proceedings of... pp 357-370
 - The workaday world as a paradigm for CSCW design
T. P. Moran, R. J. Anderson
Xerox EuroPARC
Proceedings of... pp 381-395

Desuden kan følgende bog om emnet anbefales:

- Shared Minds
The new technologies of collaboration
Michael Schrage,
Random House, 1990

jsl [SAM]

6. Opfordring

VOYVOV er et internt teknisk tidsskrift for DDE. I er alle særdeles velkomne til at sende mig indlæg til bladet. Indlæg på magnetisk form foretrækkes - brug fx en diskette, eller send mig artiklen med elektronisk post (postadresse ct).

ct [BP]