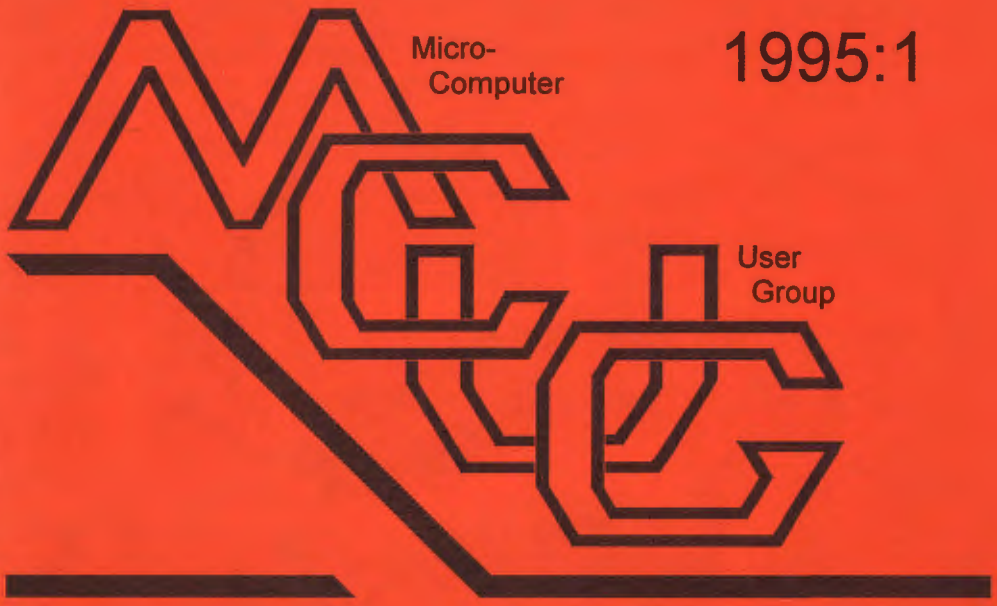


1995:1

MØDER

Der er planlagt møder følgende datoer:

mandag	23 jan	1995	Interface, PCI-bus, Hardware mv. Standarder, SSD Solid State Disks SCSI i/ii/Fast/Wide / Trends???
tirsdag	28 feb	1995	Kryptering, Public Key, Datafil DES algoritme, symmetrisk-usymm.
onsdag	22 mar	1995	Ping, Internet, e-mail og News
mandag	24 apr	1995	Åben, endnu ikke fastlagt (reserver datoen)

··· Bidrag/Indlæg/Forslag til møder og blad er velkomne! ···

Har du en god ide, et praktisk tip, evt. et forslag til noget som du gerne vil vide mere om - så lad os høre nærmere.

Møderne afholdes, hvis ikke andet er anført, på adressen:

Vesterbro Kulturhus, Lyrskovsgade 4, København V

Kl. 19. Lokale_nummer fremgår af opslag ved indgangen.

Prøv MCUG's BBS, nummeret er:

Parametre: **3160.5319**

300 - 14400 bits/sec V32bis, V42bis, MNP5

8 bit, 1 stop, ingen paritet

INDMELDELSE OG KONTINGENT

Indmeldelse i MUG Denmark foretages ved indbetaling af kontingent (225 kr. årligt) på Giro 5 686 512, MCUG Denmark /v Lars Gråbæk

Oplag: 200
Tryk: Dansk Tidsskrifts Tryk.
Redaktion: Viggo Jørgensen.
Redaktionen afsluttet primo jan 1995.

---------- RTCLOCK ----------

se også JUSTRTC i nr. 3, 1994

RTCLOCK.195

Kort beskrivelse:

RTCLOCK kan bruges til at sætte DATO og TID, til at sammenligne DOS-uret og RTC-uret, og til at synkronisere DOS-uret med RTC-uret.

Programmet er public-domain :-).

Baggrund:

De første PC'ere havde ikke noget indbygget cmos-ur. Derfor startede PC'en med at man skulle indtaste dato og tid.

System timeren er tilsluttet en frekvens på 1,193180 MHz, som neddeles med 2^16 = 65536 til ca. 18,2 afbrydelser/sekund.

De 1,193180 Mhz er f.eks. neddelt med 4 fra 4,772720Mhz, som jo netop var CPU-frekvensen på de første XT/PC'ere.

AT-bus bruger ofte en frekvens på ca. 7,159 Mhz, som neddelt med 6 også giver 1,193180 Mhz.

De ca. 18,2 afbydelser pr. sekund benyttes til at optælle en "system counter" som er et 32 bit heltal. Når tælleren efter et døgn når 60*60*24*18,206 = 1573040, nulstilles tælleren, og der markeres at man har skiftet døgn.

I nogle af de tidlige versioner af DOS, og ved brug af enkelte specielle programmer omkring midnat, risikerer man at datoen ikke bliver talt op !!

Se i RTCPROCS.C efter funktionerne "ticks2clock()" og "clock2tick()", for omregning af tid fra/til "system-counter".

Senere fik PC'erne et CMOS ur, bestående af Motorola-kredsen MC146818, eller chip/chipset kompatibel med denne.

Nu opstår så problemerne:

1) CMOS uret, herefter kaldet RTC-ur (Real-Time-Clock), benytter en krystal på 32768 Hz.

2) System uret, herefter kaldet DOS-ur, tælles op udfra de ca. 18,2 afbrydelser pr. sekund, som neddeles fra en anden krystal. Derudover kan man måske risikere at "tabe" afbrydelser, hvorved uret så også kommer bagefter.

3) Normalt stilles DOS-uret efter RTC-uret KUN ved opstart.

Højst sandsynlig vil DOS-uret afvige fra RTC-uret allerede efter nogle timer, med så meget som 10-60 sekunder.

Heldigvis er de fleste PC'ere ikke tændt i døgndrift, så DOS-uret vil være sat korrekt næste gang man tænder/boot'er.

Men for PC'ere, der skal være tændt kontinuerligt, f.eks. ved BBS drift, kan 20-30 sekunder om dagen, hurtigt blive til 2-4 minutter om ugen.
Dette er ikke heldigt, specielt hvis man har et "poll"-vindue hos andre BBS'er.

Derfor dette program til at vise både DOS-urets og RTC-urets dato og tid, og til at synkronisere DOS-uret med RTC-uret.

Funktion:

Programmet har følgende funktioner:

1) Viser både DOS-urets og RTC-urets tid.

2) Synkroniserer DOS-uret med RTC-uret.

3) Sætter dato og/eller tid kun for DOS-ur.

4a) Sætter dato og/eller tid for både DOS-ur og RTC-ur.

4b) Sætter dato og/eller tid kun for RTC-ur. Dette er kun muligt for IBM XT med BIOS efter 1/10-1986, eller PS/1 og PS/2.

5) Til test, viser gentagne gange både DOS- og RTC- tiden.

Evt. tilpasning til hardware/bios:

Idet der på visse PC'ere kan være lidt afvigelse i hvordan og hvor hurtigt opdateringen af tiden sker, er der tilføjet mulighed for at korrigere DOS-uret, ved at sætte hundreddele af sekunder til NN (=hsync). Der kan nemlig i praksis gå lidt tid fra skiftet i RTC-uret detekteres til DOS-uret sættes.

Fremgangsmåde for at finde en egnet værdi:

1) udfør: rtclock -r -x10 -y -nNN

3) prøv først med NN=0

2) se om DOS uret afviger mindre end ± 6/100 sekunder fra RTC-uret.

3) hvis ikke prøv at øge NN (<100). Oftest er værdier på ca. 5-15 passende.

Brug:

Programmet kaldes :

RTCLOCK.EXE [-?] [-r] [-s] [-h] [-d [yyyy.mm.dd]] [-t [hh:mm[:ss]]]

-? : giver en kort hjælpetekst på engelsk. (vist nedenfor)

-r : synkroniserer DOS uret med RTC uret. Ungå at kalde omkring midnat.

-s : Stiller KUN DOS-uret. Kræver enten [-d ...] og/eller [-t ...]

-c : Stiller KUN RTC-uret. Kræver enten [-d ...] og/eller [-t ...]

Hvis ingen af -c eller -s benyttes så sættes dato/tid for både DOS- og RTC-uret.

-nN : sætter DOS-ur til NN/100 sekunder ved synkronisering med "-r"

-q : "quiet", ingen uddata vises.

-p : TIL IBM PS/2. Dette vil kun virke for IBM XT m. BIOS efter 1/10-1986, eller PS/1 og PS/2.

Kort hjælpetekst:

```
>>rtclock  vers. 0.92 (1994)<<
    HELP :
    usage: RTCLOCK.EXE [-?] [-r] [-s] [-h] [-d [yyyy.mm.dd]] [-t [hh:mm[:ss]]]

    -?  : this help text
    -r  : copy hardware clock to software clock, avoid 23:59:59
    -s  : only set software clock
    -c  : only set hardware clock
    -q  : quiet-mode, no screen output
   -nN : when doing sync. "-r", set hundredths of seconds to N
    -b  : report bios date and model info.
    -p  : only works for bios: XT's after 1986-10-1, & PS/1 & PS/2.
          => uses alternate call to set DOS date, that doesn't set RTC date
          default is to set both clocks
          "/" can be used insted of "-"
```

Praktiske detaljer om programmet:

Nyeste version af programmet (RTCLKxx.ZIP) kan hentes hos

Micro Computer User Group's BBS,
tlf. 3160.5319, fidonet 2:235/15.

Programmør:

Frank Damgaard
internet frank@diku.dk
fidonet 2:235/15.2

P.S. Kommentarer/forbedringer er velkomne med e-mail til en af ovenstående adresser.

Ved en nylig afholdt VisualBasic konkurrence i København vandt Claus Lysholm, "DM"-mesterskabet + rejse til USA til en værdi af 20.000 kr. med chance for at vinde endnu mere i USA!

OS/2 Warp er den nye OS/2 version fra IBM

T-WARP2.195

Warp eller version 3 er det sidste nye OS/2 udspil fra IBM. Installationsprogrammet er blevet shinet meget op med flotte farver og en ny mulighed for avanceret eller nem installation a la Windows.

I modsætning til version 2.1 genkendes det meste hardware og de rigtige drivere installeres smertefrit uden at brugeren behøver at rette i CONFIG.SYS.

Jeg har kørt med Warp'en i nogle måneder nu og stabiliteten er helt ok.

Hastigheden skulle være forbedret, deraf navnet Warp som jo antyder lyshastighed, men jeg syntes nu ikke Warp'en kører hurtigere end 2.11.

Har man lydkort i PC'en, høres ved start af OS/2 den lyd fra science-fiction film, der markerer overgang til Warp speed.

Af nye ting er der kommet en *undo arrange* feature, således at hvis man får arrangeret sine ikoner forkert på Desktop'en, så kan man undo'e og vupti har man sin gamle Desktop igen. Ikonerne er blevet flottere med 3D udseende.

Der er mulighed for at tage backup i 3 generationer af ens systemfiler samt Desktop. (Er der nogen, der ikke véd det er Desktop'en altså OS/2 grafiske brugergrænseflade a la Windows).

De små uduelige programmer i *"productivity"* folderen er erstattet af en såkaldt Bonuspakke med integreret tekstbehandling, regneark og database.

Dertil er der et væld af programmer til at kommunikere med Internet, men for at blive registreret, skal man have et kreditkort: VISA, MasterCard eller andet internationalt anerkendt kreditkort.

Jeg har ikke leget meget med Bonuspakken da jeg stadig bruger mine gamle favorit DOS programmer. Og det er OS/2 jo helt cool overfor, så jeg sidder nu og taster ubesværet i DOS programmet WordpPerfect 5.1 samtidig med at jeg tager en total backup på DAT bånd af mine harddiske.

OS/2 er perfekt til multitasking.

I Bonuspakken findes et systemutility værktøj, der fortæller alt om maskinens indre. Det siger, at OS/2's interne versionsnr. er 2.3 og det er nok mere rimeligt end 3.0, for så meget er der altså ikke sket med OS/2 siden 2.11.

En anden ny smart ting i Warp'en er at man kan sætte OS/2 op til at lukke *"forældre"* vinduer, når man klikker sig vej i hierarkiet af foldere for at finde og køre et bestemt program, som jo på Desktop'en er repræsenteret ved en ikon ligesom i Windows.

En tredje nyhed er den på engelsk kaldte *"launchpad"*, som er et WPS objekt, der af brugeren nemt kan sættes op til at give hurtig adgang til ofte brugte programmer.

Man må håbe, at OS/2 med den nye Warp nu slår igennem, OS/2 er og har længe været så godt et system, at det fortjener stor udbredelse.

John B. Jacobsen.

OS/2 WARP, pt. alm. priser incl. moms.

OS/2 WARP Dansk eller Engelsk på 3,5"		857 kr.
OS/2 WARP Dansk eller Engelsk på 3,5"	opgr.	513 kr.
OS/2 WARP Dansk eller Engelsk på CD		772 kr.
OS/2 WARP Dansk eller Engelsk på CD	opgr.	426 kr.

RAR - nok et pakkeprogram til DOS!!

RAR-ART.195

Jeg har netop haft fornøjelsen af at gennemføre en ultra kort test af et nyt shareware pakkeprogram med rødder i de dybe finske skove eller måske endda endnu længere østpå.

Det første spørgsmål der trænger sig på er selvfølgelig hvorfor endnu et pakkeprogram? Og der må jeg nok sige at der bliver jeg svar skyldig! Når jeg alligevel giver RAR et par ord med på vejen, så er det fordi det har et helt andet bruger-interface end nogen af de andre pakkeprogrammer jeg er stødt på.

RAR kommer som en lille selvudpakkende fil på ca. 140 kB, efter udpakningen har man den exekverbare fil (RAR.EXE), de fornødne dokumentationsfiler og et lille RAR arkiv med en lille stand alone udpakker, disse filer fylder omkring 350 kB.

Så er man faktisk køreklar, programmet kan enten køre fuldstændig kommandolinie orienteret som andre pakkere; men den kan også køres i en fuld-skærms mode, hvor skærm billede og betjeningsmåde minder meget om det, der kendes fra NORTON COMMANDER.

I denne mode får man vist en filliste med filerne i det aktuelle bibliotek og kan så med <ins> udpege de filer man vil have med i sit arkiv, man kan også vælge ved hjælp af wildcards ved at bruge <+>'et på det numeriske tastatur.

Således kan man hurtigt og overskueligt udvælge de filer man ønsker at få med i sit arkiv. Selve pakningen startes så ved hjælp af <f2>, her spørges først om et arkiv navn og derefter går resten af sig selv. Der kan her gives både et nyt arkiv navn eller navnet på et eksisterende arkiv, hvis der skal tilføjes filer til et eksisterende arkiv.

Hvis markøren står på et RAR, ZIP, ARJ eller LZH arkiv i fillisten får man, ved at trykke <ENTER>, en ny filliste med de filer der ligger i det pågældende arkiv. Peger man nu på en af de arkiverede filer kan man med den indbyggede viewer se indholdet af den pågældende fil.

Der er i RAR forskellige konfigurerings muligheder, man kan f. eks. vælge mellem 6 kompressions metoder, verificering, password beskyttelse mv. Man kan også vælge mellem noget der kaldes 'solide' arkiver og normale arkiver. Det er ikke lykkedes mig at gennemskue den præcise forskel; men jeg mener forskellen ligger i hvordan man adskiller filerne fra hinanden i arkiverne. I de 'solide' arkiver pakkes der helt tæt, så der kræves mere detaljeret analyse for at finde ud af hvor de enkelte filer adskilles.

Solide arkiver er derfor langsomme, hvis man bruger dem til arkiver der ofte skal modificeres, til gengæld er de pladsbesparende.

I min mini test har jeg ikke set på tidsforbruget; men kun på kompressionen, jeg lod RAR (1.52) og PKZIP (2.04g) pakke de samme 13 filer (12 små ASCII filer og en EXE fil) ialt 225 693 bytes, begge med deres standard kompressions metode og i normale arkiver, de resulterende filer havde følgende størrelser:

TEST.RAR 85 444 bytes
TEST.ZIP 88 106 bytes.

Så i dette tilfælde pakkede RAR altså bedre end PKZIP, jeg vil dog være forsigtig med at uddrage nogen konklusion af dette, da jeg ikke har lavet andre sammenligninger (flere store filer, filer af andre typer osv.)

Uden at have høstet de store erfaringer med denne pakker må jeg sige at den har været et yderst charmerende bekendtskab, så den fortjener yderligere undersøgelse!

LG.

Programmet hedder RAR, den version jeg har kikket på er 1.52. Det er share ware, registrering koster 35$.

I dette nummer af bladet er medtaget en stor del materiale vedr. kryptering / de-kryptering. Det er et område af databehandling, som får større og større betydning, ikke kun for militær og diplomati, men også erhvervsliv og - privatliv (pin-koder fx.). Selvom sidstnævnte måske er lidt søgt, så er det dog et forsøg på hemmeligholdelse af nogle cifre!

Af hensyn til de fleste af os, som ikke er prof. kryptografer, men synes at emnet er interessant, startes lidt blødt op, så læseren kan nå at få sjælen med til de noget tungere ting. Og *huskat* d. **28 feb 95, kl. 19** har vi et interessant medlemsmøde om netop dette emne!

(red.)

Kryptering: Fra Cæsar til EF

Reprint fra EC-NYT. (Reprint fra MCUG 1992:4)

Kryptering er kort fortalt en metode til at gøre informationer uforståelige for uvedkommende, men således at den rette modtager er i stand til at genskabe informationen.

Der er tale om en matematisk baseret teknik til at hemmeligholde et budskab, og kun personer med kendskab til den rette nøgle vil være i stand til at læse budskabet.

Der er to problemer i forbindelse med anvendelse af krypteringsteknikker. Det første er valget af selve krypteringsalgoritmen. Herpå findes der i dag udmærkede løsninger.

Det andet problem er fastlæggelse af kommunikationsprotokollen. Dette problem kan måske synes lettere end det første, men i praksis viser det sig ofte at være sværere at håndtere.

I det følgende vil vi se lidt nøjere på krypteringsalgoritmer.

Krypteringsteknikker.

Der er i tidens løb udviklet mange forskellige krypteringsalgoritmer. De fleste kan dog betragtes som varianter af to klassiske metoder.

Den første metode kaldes substitutionsmetoden. Metoden har været anvendt helt tilbage til det gamle Rom og tilskrives ofte Julius Cæsar, der åbenbart ikke følte, at han kunne stole på sine nærmeste.

Krypteringsforskriften ved substitutionsmetoden fremkommer ved at opskrive alfabetet 2 gange under hinanden, således at det nederste alfabet er forskudt i forhold til det øverste:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ
BCDEFGHIJKLMNOPQRSTUVWXYZÆØÅA
```

Nøglen til krypteringen er antallet af pladser, som alfabeterne er forskudt i forhold til hinanden.

I dette tilfælde er forskydningen 1 plads, dvs. A bliver til B, B bliver til C osv.

For eks. bliver

```
ELEKTRONIK  til
FMFLUSPOJL,
```

hvilket umiddelbart er ganske uforståeligt.

Substitutionsmetoden er set benyttet ved prismærkning af tøj, hvor indkøbsprisen er krypteret, medens salgsprisen står med klare tal. En anden klassisk teknik kaldes permutationsmetoden. Ved denne metode opskrives de enkelte bogstaver fra den oprindelige tekst i en anden rækkefølge efter en nøje fastlagt forskrift.

For eks. kan algoritmen være at ombytte bogstaverne parvis.

Herved bliver

```
ELEKTRONIK    til
LEKERTNOKI,
```

hvilket heller ikke umiddelbart kan forstås.

Disse to metoder er sikkert velkendte for de

fleste. De anvendes ofte af skolebørn, når hemmeligheder sendes rundt i klassen.
Dette kan give anledning til søvnløse nætter for nysgerrige lærere.

Det følgende eksempel viser samme tekst, først krypteret efter substitutionsmetoden og dernæst efter permutationsmetoden.

```
BCRC PKCL LCQI CJGE RØRD CHJC
TDEE ERNM SNKE IEGL TTFA LEEJ
```

Der er en principiel forskel på de to metoder. Substitutionsmetoden er en løbende kryptering, medens permutationsmetoden er en blok-kryptering, dvs. teksten skal deles op i blokke, før krypteringén kan foretages.
Som et eksempel på en moderne krypteringsalgoritme, der bygger på de klassiske metoder, kan nævnes DES.
DES er en forkortelse for Data Encryption Standard. Algoritmen er udviklet af IBM og anbefales i øjeblikket af de amerikanske myndigheder til brug af både private virksomheder og offentlige institutioner.
Krypteringsmetoden i DES består af en række skiftevise permutationer og substitutioner. Selv om de to teknikker hver for sig ikke er særlig effektive, viser det sig, at en passende sammensætning giver en ganske effektiv kryptering.
Dette illustreres af, at der på trods af talrige analyser i fagtidsskrifter endnu ikke er offentliggjort en effektiv metode til at bryde DES krypteringsalgoritmen.
DES er dog til stadighed genstand for megen diskussion, og det forventes, at de amerikanske anbefalinger ændres inden for en kortere årrække.

EC og datasikkerhed.

Elektronikcentralen deltager i øjeblikket i 2 EF-projekter vedrørende beskyttelse af data og programmel.
Projekterne gennemføres i samarbejde med Agence de l'Informatique i Paris,
The National Computing Centre i Manchester og National Software Centre i Dublin.
De to projekter omhandler henholdsvis netværkssikkerhed og programmelintegritet.
Netværkssikkerhed er blevet aktuel for en stor gruppe virksomheder som følge af den teknologiske udvikling på telekommunikationsområdet.for informationsudveksling er der opstået en række problemer.
Hurtig adgang til præcise informationer er livsnerven for mange virksomheder, men samtidig et sårbart område, hvis informationerne misbruges eller forvanskes.
Virksomhederne må indstille sig på at leve med begreber som sikkerhedsprogrammel, krypteringsapparater, identifikationssystemer, autentifikations- og signatursystemer.

Programmelintegritet - altså at programmerne er korrekte, pålidelige og troværdige, og at disse egenskaber bevares - kan også være et sikkerhedskritisk område for mange virksomheder. Integritet er derfor en egenskab, der skal indbygges i programmerne allerede i udviklingsfasen.
Formålet med de to projekter er at undersøge markedet for relevante produkter og teknikker, samt vurdere disses effektivitet og økonomi, sammenholdt med forskellige brugergruppers behov.
Der vil blive udarbejdet anbefalinger til computerproducenter, systemdesignere og brugere. Hermed vil forskellige brugergrupper, med hver sine specielle krav, kunne finde oplysninger om, hvilke realistiske muligheder, der eksisterer for at sikre kommunikation og programmer.
Projekterne har således interesse for en meget bred kreds, og foreløbig har en stor computerproducent indledt et samarbejde med projektgruppen.

Jørgen Bøegh.

PS: Den krypterede tekst lyder på dansk:
»Det er menneskeligt at fejle«

Hvis Operativsystemer havde været Luftfartsselskaber

OPSYSFLY.195

En lørdag lå det nyeste nummer af tidsskriftet Layout i min postkasse. Deri fandt jeg en lille artikel, som klart adskilte sig fra de øvrige ved at sammenligne Operativsystemer med Luftfartsselskaber.
Jeg må tilstå, at der i dette korte afsnit var et og andet som gjorde det vanskeligt at holde ansigtet i alvorlige folder hele tiden. - Derfor skal jeg ikke forholde læserne indsigt i artiklens indhold.

DOS Airlines
Alle skubber flyet igang, indtil det letter. Så springer alle ombord og lader flyet komme i spin, indtil det igen rammer jorden.
Herefter skubber alle igen, springer ombord...

DOS med Quemm Airlines
Nøjagtig som DOS Airlines, blot med mere plads til fødderne ved start-skubning.

Mac Airlines
Alle stewarder, stewardesser, piloter, jordpersonel og billetsælgere ser ens ud, bevæger sig ens og siger det samme.
Når man spørger om detailler, får man altid det samme svar:
Det må man ikke vide. Vil ikke vide det, og alt fungerer helt rigtigt. Man skal være helt stille.

Windows Airlines
Lufthavnsterminalen har smukke kulører, stewarder og stewardesser er venlige.
Man når uden problemer ombord, en helt fejlfri start...
Pludselig styrter flyet ned, uden nogen som helst form for advarsel.

OS/2 Airlines
For at komme ombord på flyet, skal man have sin billet stemplet 10 gange og stillet sig i 10 forskellige køer.
Så udfylder man en formular, i hvilken man skal anføre, hvor man ønsker at sidde og om siddepladsen skal ligne en på et skib, i et tog eller i en bus.
Når det endelig er lykkedes at komme ombord og når flyet virkelig hæver sig fra jorden, har man en vidunderlig flyvning... bortset fra når højde- og sideror dækkes af overisning. I så tilfælde har man i det mindste altid tilstrækkelig tid til at forberede sig på en nedstyrtning.

Unix Airlines
Enhver, som vil ombord, medbringer et stykke af flyet til lufthavnen.
Alle går ud på startbanen og sætter flyet sammen stykke for stykke.
Under udførelsen af dette diskuterer man fortløbende, hvilken slags fly man netop er i færd med at samle.

NT Airlines
Alle går ud på startbanen, fremsiger i kor password og danner et omrids af en flyvemaskine. Så sætter de sig på jorden og afgiver støj og lugt, som om de virkelig ville flyve. Derefter går de med en god følelse hjem.

Jeg håber, at I også har haft en smule fornøjelse heraf :-) *JK.*

Krypterings-algoritmen GOST contra DES

Bruce Schneier.
Dr. Dobb's Journal, januar 1995.

(Data Encryption Standard)

GOSTDK.195

Som kryptograf bliver jeg sommetider præsenteret for virkeligt spændende teknologi. Sidste år blev jeg f. eks. introduceret til GOST, mit første møde med krypterings algoritmer fra det tidligere Sovjetunionen. Algoritmen, hvis formelle navn er:
"Cryptographic Transformation Algorithm-GOST 28147-89", blev publiceret af Sovjetunionens nationale standardiseringskontor (kendt under forkortelsen "GOST").
Algoritmen var ikke klassificeret, hvilket antyder at den skulle anvendes til beskyttelse af civile, ikke miltære, data. Spørg mig ikke hvordan algoritmen slap ud af landet.
Her vil jeg beskrive GOST og dens relationer til DES samt undersøge sikkerheden ved brug af GOST algoritmen. GOST er stadig ny for kryptografer fra vesten, så mere detaljeret information om sikkerheden ved GOST vil fremkomme i takt med at folk får en chance for at studere den.

En beskrivelse af GOST

GOST er en algoritme med hemmelig nøgle, med en konstruktion der minder om DES's. Jeg tror den blev opfundet af sovjetiske kryptografer der havde studeret DES. Offentliggørelsen i 1989 som sovjetisk standard kom nogen tid efter at DES var blevet offentlig tilgængelig; men det er ikke muligt at vide hvornår den er udviklet og hvor længe den blev brugt indenfor det sovjetiske statsapparat.
GOST er ligesom DES et Feistel netværk, algoritmen gentager en simpel krypterings algoritme i flere runder. Teksten brydes først op i en venstre halvdel, *L*, og en højre halvdel, *R*.
K er del-nøglen for den givne runde.
En runde, *i*, af hver algoritme ser ud som følger:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \text{ XOR } f(R_{i-1}, K_i)$$

DES har 16 runder medens GOST har 32. Begge algoritmer kan nemt vendes, således at afkryptering er det samme som kryptering bare med *K'*erne brugt i omvendt rækkefølge.

Figur 1 viser en enkelt runde af GOST. Først lægges højre halvdel og den $i'te$ delnøgle sammen modulo 2^{32} (det er det samme som en 32-bit addition, hvor man ikke tager hensyn til en eventuel mente af højeste orden).
Resultatet brydes så op i otte 4-bit stumper, der hver bruges som input til forskellige S-bokse.
Der er otte forskellige S-bokse i GOST; de første fire bit puttes i den første S-boks, de næste fire bit i den næste o.s.v.

Hver S-boks er en permutation[1]) af numrene 0 til 15, en S-boks kunne f.eks se således ud: *7, 10, 2, 4, 15, 9, 0, 3, 6, 12, 5, 13, 1, 8, 11, 14.* Hvis input til denne S-boks er 0 bliver resultatet 7, hvis input er 1 bliver resultatet 13 o.s.v.
Alle otte S-bokse er forskellige og skal betragtes som dele af kode-nøglen.
S-boksene skal holdes hemmelige.

Resultaterne fra de otte S-bokse kombineres til et nyt 32 bit ord, hvorefter hele ordet underkastes en cirkulær rokade 11-bit mod venstre (mod højere ordens bit'ene).
Endelig adderes resultatet modulo 2^{32} til den venstre halvdel for at blive den nye højre halvdel og den højre halvdel bliver den nye venstre halvdel.
Det gentages 32 gange, så er man færdig.

Del-nøglerne genereres meget enkelt, 256-bit nøglen deles i otte 32-bit blokke. Den første blok bruges i første runde, anden blok i anden runde o.s.v.

---

[1]) rækkefølge-ændring

I den 9. runde og igen i den 17. runde starter sekvensen igen, således at den første blok bruges i den 9. og den 17 runde, den anden blok bruges i den 10. og den 18. runde o.s.v. I den 25. til den 32. runde tages blokkene i omvendt rækkefølge, således at den ottende blok bruges i den 25. runde, den syvende blok bruges i den 26. runde, og den sjette blok bruges i den 27. runde o.s.v.

Sammenligning med DES

Til sammenligning viser figur 2 en enkelt runde af DES (DES starter og slutter med en permutation, dette øger ikke algoritmens sikkerhed og er helt udeladt i denne diskussion).

Først udvides højre halvdel fra 32 til 48 bits ved en fast permutation.

Resultatet XOR'es med den *i'te* delnøgle og brydes så op i otte 6-bit stykker.

Disse stykker bruges som input til hver sin S-boks. DES har 8 forskellige S-bokse; de første 6 bit kommer i den første S-boks, de næste 6 bit i den anden S-boks o.s.v.

Hver S-boks er fire permutationer af numrene 0 til 15. I DES holdes S-boksene fast og de er offentlige, de er simpelthen en del af standarden og er kendt af alle.

Resultaterne fra de otte S-bokse kombineres til et 32-bit ord, hvorefter hele ordet permuteres med en anden bestemt permutation. Endelig adderes resultatet modulo 2^{32} til venstre halvdel og bliver den nye højre halvdel og den højre halvdel bliver den nye venstre halvdel, dette gentages 16 gange.

Proceduren til at generere DES's delnøgler er kompliceret. 56-bit nøglen deles først i to lige store dele.

Hver 28-bit del rokeres nu 1 eller 2 bit cirkulært mod venstre afhængigt af runden. Efter denne rokade vælges 48 bits efter en fast permutation.

De væsentligste forskelle mellem DES og GOST kan sammenfattes således:

- DES har en kompliceret procedure til at generere del-nøgler fra nøglerne, hvorimod GOST har en meget simpel procedure.
- DES har en 56-bit nøgle, GOST har en 256-bit nøgle. Hvis du dertil lægger informationen i de hemmelige S-bokse har GOST i alt ca. 610 bits hemmelig information.
- S-boksene i DES har 6-bit input og 4-bit output. S-boksene i GOST har 4-bit input og 4- bit output.
- Begge algoritmer har 8 S-bokse; men en S-boks i GOST er fire gange mindre end en S-boks i DES.
- DES har en irregulær permutation kaldet en "P-boks", GOST bruger et 11-bit cirkulært skift.
- DES bruger XOR for at addere nøglen til den højre halvdel og for at addere den højre halvdel til den venstre halvdel; GOST bruger addition modulo 2^{32} for begge disse operationer.
- DES har 16 runder, GOST har 32.

Generering af S-bokse til GOST

GOST standarden diskuterer ikke hvordan man genererer S-boksene, de leveres bare på en eller anden måde. Nylige opdagelser indenfor differential- og lineær kryptografi analyse viser at valget af S-bokse har stor betydning for den opnåede sikkerhed, sagt på en anden måde, der er gode S- bokse og der er dårlige S-bokse. Det gælder således også for GOST, det har ført til spekulationer om hvorvidt det tidligere sovjetstyre forsynede organisationer der stod sig godt med styret med gode S-bokse, hvorimod organisationer som styret ønskede at aflure fik dårlige S-bokse. Det kan selvfølgelig være rigtigt; men yderligere kontakt med en russisk GOST chip producent giver andre mulige alternativer.

Han genererede S-boks permutationerne selv ved hjælp af en tilfældig-tals generator. S-boksene i den medfølgende kildetekst blev brugt i et sovjetisk bank område.

Er sikkerheden god?

Er GOST en sikker algoritme? Det korte svar er, at det er der ikke nogen der ved.

Her er et lidt længere svar:

Hvis vi antager at GOST algoritmen kun kan brydes ved "rå vold", så er den klart mere sikker end DES. GOST har en 256-bit nøgle – ja endnu længere hvis du tæller de hemmelige S-bokse med.

Overfor differential- og lineær kryptografi analyse er GOST måske mere sikker selvom de tilfældigt genererede S-bokse i GOST muligvis er svagere end de fast valgte S-bokse i DES, til gengæld er de en del af nøglen. Hemmeligholdelsen af S-boksene i GOST øger dens sikkerhed overfor systematiske (differential- og lineær analyse) angreb.

For begge typer angreb gælder at angrebet bliver vanskeligere jo flere runder algoritmen indeholder.

GOST har dobbelt så mange runder som DES og det umuliggør formodentlig både lineære- og differentiale angreb.

For de andre elementers vedkommende står de to algoritmer lige eller GOST står svagere.

GOST har ikke den samme ekspansions permutation som DES har. Det vides at hvis man fjerner denne permutation vil DES blive svagere, det reducerer den såkaldte lavine effekt; (den effekt som forandring af en enkelt bit i input har på bits i outputtet).

Det er rimeligt at tro, at GOST er svagere fordi den ikke har denne permutation.

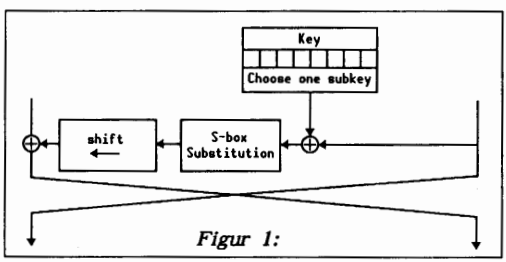
At GOST bruger addition i stedet for XOR gør formodentlig ingen forskel.

Den mest alvorlige ændring ser ud til at være efter S-boksene: GOST's brug af et cyclisk skift i stedet for en permutation.

DES permutationen er designet til at øge lavine effekten:

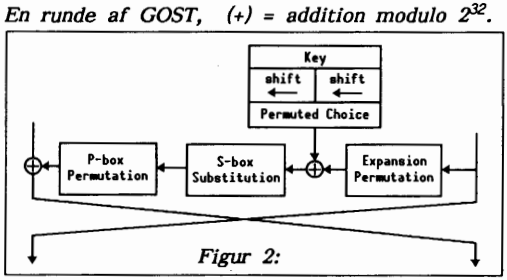
I GOST påvirker en ændring af een bit i input een S-boks i een runde, som derefter påvirker to S-bokse i den næste runde, tre i runden derefter o.s.v. Der går således otte runder før en forandring af en enkelt input bit i GOST påvirker alle output bits;

i DES kræves der kun fem runder. Det er oplagt en svaghed ved GOST; men husk at GOST har 32 runder hvor DES har 16.



Figur 1:

En runde af GOST, (+) = addition modulo 2^{32}.



Figur 2:

En runde af DES, (+) = exklusiv-OR.

Konklusion

GOST's designere har prøvet at opnå balance mellem effektivitet og sikkerhed.

De fulgte det grundlæggende design i DES, men lavede nogle modifikationer til algoritmen.

Som Listning 1 viser er resultatet en algoritme der er velegnet til implementering i software (der er ingen irriterende bit ombytninger).

De synes at have været usikre på sikkerheden i deres algoritme, det har de så prøvet at kompensere for ved at anvende en meget lang nøgle, ved at hemmeligholde S-boksene og endelig ved at fordoble antallet af runder.

Om deres anstrengelser har ført til en algoritme, der er mere sikker end DES kan ikke afgøres endnu.

Fordansket af *Lars Gråbæk.*

GOST - Listing 1

(modifications: E. Sønderhousen)

```
/* The GOST 28147-89 cipher, by Colin Plumb */    /* GOST2LST.CN, mcug */

/* If you read the standard, it belabors the point of copying correspon-
 * ding bits from point A to point B quite a bit.  It helps to understand
 * that the standard is uniformly little-endian, although it numbers bits
 * from 1 rather than 0, so bit n has value 2^(n-1).  The least signifi-
 * cant bit of the 32-bit words that are manipulated in the algorithm is
 * the first, lowest-numbered, in the bit string.
 */
#define TEST                        /* added */
                                    /* A 32-bit data type */
#ifdef __alpha                      /* Any other 64-bit machines? */
typedef unsigned int word32;
#else
typedef unsigned long word32;
#endif

static unsigned char const k8[16] = {
     1, 15, 13,  0,  5,  7, 10,  4,  9,  2,  3, 14,  6, 11,  8, 12 };
static unsigned char const k7[16] = {
    13, 11,  4,  1,  3, 15,  5,  9,  0, 10, 14,  7,  6,  8,  2, 12 };
static unsigned char const k6[16] = {
     4, 11, 10,  0,  7,  2,  1, 13,  3,  6,  8,  5,  9, 12, 15, 14 };
static unsigned char const k5[16] = {
     6, 12,  7,  1,  5, 15, 13,  8,  4, 10,  9, 14,  0,  3, 11,  2 };
static unsigned char const k4[16] = {
     7, 13, 10,  1,  0,  8,  9, 15, 14,  4,  6, 12, 11,  2,  5,  3 };
static unsigned char const k3[16] = {
     5,  8,  1, 13, 10,  3,  4,  2, 14, 15, 12,  7,  6,  0,  9, 11 };
static unsigned char const k2[16] = {
    14, 11,  4, 12,  6, 13, 15, 10,  2,  3,  8,  1,  0,  7,  5,  9 };
static unsigned char const k1[16] = {
     4, 10,  9,  2, 13,  8,  0, 14,  6, 11,  1, 12,  7, 15,  5,  3 };

/* Byte-at-a-time substitution boxes */
static unsigned char k87[256];
static unsigned char k65[256];
static unsigned char k43[256];
static unsigned char k21[256];

/* Build byte-at-a-time substitution tables.
 * This must be called once for global setup
 */
void
kboxinit(void)
{
    int i;
    for (i = 0; i < 256; i++) {
        k87[i] = k8[i >> 4] << 4 | k7[i & 15];
        k65[i] = k6[i >> 4] << 4 | k5[i & 15];
        k43[i] = k4[i >> 4] << 4 | k3[i & 15];
        k21[i] = k2[i >> 4] << 4 | k1[i & 15];
    }
}

/* Do the substitution and rotation that are the core of the operation,
 * like the expansion, substitution and permutation of the DES.
 * It would be possible to perform DES-like optimisations and store the
 * table entries as 32-bit words, already rotated, but the efficiency
 * gain is questionable. This should be inlined for maximum speed
 */
#if __GNUC__
  inline __
#endif
static word32
f(word32 x)
{
        /* Do substitutions */
#if 0
        /* This is annoyingly slow */
    x = k8[x>>28 & 15] << 28 | k7[x>>24 & 15] << 24 |
        k6[x>>20 & 15] << 20 | k5[x>>16 & 15] << 16 |
        k4[x>>12 & 15] << 12 | k3[x>> 8 & 15] <<  8 |
        k2[x>> 4 & 15] <<  4 | k1[x     & 15];
#else
        /* This is faster */
    x = k87[x>>24 & 255] << 24 | k65[x>>16 & 255] << 16 |
        k43[x>> 8 & 255] <<  8 | k21[x  & 255];
#endif

        /* Rotate left 11 bits */
    return x<<11 | x>>(32-11);
}

/* The GOST standard defines the input in terms of bits 1..64, with
 * bit 1 being the lsb of in[0] and bit 64 being the msb of in[1].
 * The keys are defined similarly, with bit 256 being the msb of key[7].
 */
void
gostcrypt(word32 const in[2], word32 out[2], word32 const key[8])
{
    register word32 n1, n2; /* As named in the GOST */

    n1 = in[0];
    n2 = in[1];

    /* Instead of swapping halves, swap names each round */
    n2 ^= f(n1+key[0]);
    n1 ^= f(n2+key[1]);
    n2 ^= f(n1+key[2]);
    n1 ^= f(n2+key[3]);
    n2 ^= f(n1+key[4]);
    n1 ^= f(n2+key[5]);
    n2 ^= f(n1+key[6]);
    n1 ^= f(n2+key[7]);

    n2 ^= f(n1+key[0]);
    n1 ^= f(n2+key[1]);
    n2 ^= f(n1+key[2]);
    n1 ^= f(n2+key[3]);
    n2 ^= f(n1+key[4]);
    n1 ^= f(n2+key[5]);
    n2 ^= f(n1+key[6]);
    n1 ^= f(n2+key[7]);

    n2 ^= f(n1+key[0]);
```

```
    n1 ^= f(n2+key[1]);
    n2 ^= f(n1+key[2]);
    n1 ^= f(n2+key[3]);
    n2 ^= f(n1+key[4]);
    n1 ^= f(n2+key[5]);
    n2 ^= f(n1+key[6]);
    n1 ^= f(n2+key[7]);

    n2 ^= f(n1+key[7]);
    n1 ^= f(n2+key[6]);
    n2 ^= f(n1+key[5]);
    n1 ^= f(n2+key[4]);
    n2 ^= f(n1+key[3]);
    n1 ^= f(n2+key[2]);
    n2 ^= f(n1+key[1]);
    n1 ^= f(n2+key[0]);

    /* There is no swap after the last round */

    out[0] = n2;
    out[1] = n1;
}

/* The key schedule is somewhat different for decryption. (The key table is
 * used once forward and 3 times backward). You could define an expanded
 * key, or just write the code twice, as done here.
 */
void
gostdecrypt(word32 const in[2], word32 out[2], word32 const key[8])
{
    register word32 n1, n2; /* As named in the GOST */

    n1 = in[0];
    n2 = in[1];

    n2 ^= f(n1+key[0]);
    n1 ^= f(n2+key[1]);
    n2 ^= f(n1+key[2]);
    n1 ^= f(n2+key[3]);
    n2 ^= f(n1+key[4]);
    n1 ^= f(n2+key[5]);
    n2 ^= f(n1+key[6]);
    n1 ^= f(n2+key[7]);

    n2 ^= f(n1+key[7]);
    n1 ^= f(n2+key[6]);
    n2 ^= f(n1+key[5]);
    n1 ^= f(n2+key[4]);
    n2 ^= f(n1+key[3]);
    n1 ^= f(n2+key[2]);
    n2 ^= f(n1+key[1]);
    n1 ^= f(n2+key[0]);

    n2 ^= f(n1+key[7]);
    n1 ^= f(n2+key[6]);
    n2 ^= f(n1+key[5]);
    n1 ^= f(n2+key[4]);
    n2 ^= f(n1+key[3]);
    n1 ^= f(n2+key[2]);
    n2 ^= f(n1+key[1]);
    n1 ^= f(n2+key[0]);

    out[0] = n2;
    out[1] = n1;
}

/* The GOST "Output feedback" standard. It seems closer morally to the
 * counter feedback mode some people have proposed for DES.
 *
 * The IV is encrypted with the key to produce the initial counter value.
 * Then, for each output block, a constant is added, modulo 2^32-1 (0 is
 * represented as all-ones, not all-zeros), to each half of the counter,
 * and the counter is encrypted to produce the value to XOR with the
 * output.
 * Len is the number of blocks. Sub-block encryption is left as an exer-
 * cise for the user. Remember that the standard defines everything in a
 * little-endian manner, so you want to use the low bit of gamma[0] first.
 *
 * OFB is, of course, self-inverse, so there is only one function.
 */

   /* The constants for addition */
#define C1 0x01010104
#define C2 0x01010101

void
gostofb(word32 const *in, word32 *out, int len,
    word32 const iv[2], word32 const key[8])
{
   word32 temp[2];              /* Counter */
   word32 gamma[2];             /* Output XOR value */

   /* Compute starting value for counter */
   gostcrypt(iv, temp, key);

   while (len--) {
       temp[0]  += C2;
       if  (temp[0] < C2)       /* Wrap modulo 2^32? */
           temp[0]++;           /* Make it modulo 2^32-1 */
        temp[1] += C1;
        if (temp[1] < C1)       /* Wrap modulo 2^32? */
            temp[1]++;          /* Make it modulo 2^32-1 */

        gostcrypt(temp, gamma, key);

        *out++ = *in++ ^ gamma[0];
        *out++ = *in++ ^ gamma[1];
   }
}

/*
```

```
 * The CFB mode is just what you'd expect. Each block of ciphertext y[]
 * is derived from the input x[] by the following pseudocode:
 * y[i] = x[i] ^ gostcrypt(y[i-1])
 * x[i] = y[i] ^ gostcrypt(y[i-1])
 * Where y[-1] is the IV.
 *
 * The IV is modified in place.  Again, len is in *blocks*.
 */

void
gostcfbencrypt(word32 const *in, word32 *out, int len,
           word32 iv[2], word32 const key[8])
{
     while (len--) {
         gostcrypt(iv, iv, key);
         iv[0] = *out++ ^= iv[0];
         iv[1] = *out++ ^= iv[1];
     }
}
void
gostcfbdecrypt(word32 const *in, word32 *out, int len,
           word32 iv[2], word32 const key[8])
{
     word32 t;
     while (len--) {
         gostcrypt(iv, iv, key);
         t = *out;
         *out++ ^= iv[0];
         iv[0] = t;
         t = *out;
         *out++ ^= iv[1];
         iv[1] = t;
     }
}

/* The message suthetication code uses only 16 of the 32 rounds. There
 * *is* a swap after the 16th round. The last block should be padded
 * to 64 bits with zeros. len is the number of *blocks* in the input.
 */
void
gostmac(word32 const *in, int len, word32 out[2], word32 const key[8])
{
    register word32 n1, n2;         /* As named in the GOST */

    n1 = 0;
    n2 = 0;

    while (len--) {
       n1  = *in++;
       n2  = *in++;

       /* Instead of swapping halves, swap names each round */
       n2 ^= f(n1+key[0]);
       n1 ^= f(n2+key[1]);
       n2 ^= f(n1+key[2]);
       n1 ^= f(n2+key[3]);
       n2 ^= f(n1+key[4]);
       n1 ^= f(n2+key[5]);
       n2 ^= f(n1+key[6]);
       n1 ^= f(n2+key[7]);

       n2 ^= f(n1+key[0]);
       n1 ^= f(n2+key[1]);
       n2 ^= f(n1+key[2]);
       n1 ^= f(n2+key[3]);
       n2 ^= f(n1+key[4]);
       n1 ^= f(n2+key[5]);
       n2 ^= f(n1+key[6]);
       n1 ^= f(n2+key[7]);
    }
    out[0] = n1;
    out[1] = n2;
}

#ifdef TEST

#include <stdio.h>
#include <stdlib.h>

/* Designed to cope with 15-bit rand() implementation */
#define RAND32 ((word32)rand() << 17 ^ (word32)rand() << 9 ^ rand())

int
main(void)
{
     word32 key[8];
     word32 plain[2];
     word32 cipher[2];
     int i, j;

     kboxinit();

    printf("GOST 21847-89 test driver.\n");

     for (i = 0; i < 1000;  i++) {
         for (j = 0; j < 8; j++)
           key[j] = RAND32;
         plain[0] = RAND32;
         plain[1] = RAND32;

         printf("%3d\r", i);
         fflush(stdout);

         gostcrypt(plain, cipher, key);
         for (j = 0; j < 99; j++)
             gostcrypt(cipher, cipher, key);
         for (j = 0; j < 100; j++)
             gostdecrypt(cipher, cipher, key);

             if (plain[0] != cipher[0] || plain[1] != cipher[1]) {
                 fprintf(stderr, "\nError! i = %d\n", i);
                 return 1;
             }
    }
    printf("All test passed.\n");
    return 0;
}
#endif  /* TEST */
```

Opgave 3, fra sidste nummer: Ingen forslag modtaget endnu.

En artikel om **Euklid** er blevet 'skubbet' til en senere udgave af bladet.

Ny Opgave 4:

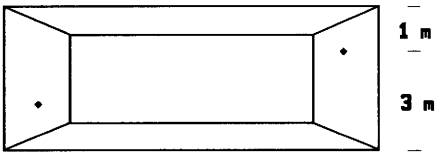
Denne lille fortælling siges at være foregået i Grækenland.

Opgaven hører ikke til de allersværeste! En meget træt og sulten bi er landet i et rum, som måler 4x4 meter (højde og bredde) og 10m lang. Midt på den ene endevæg, 1 meter oppe sidder bien;
- den har fået øje på en lille blomst med pollen i en revne midt på den modsatte endevæg, desværre 3 meter oppe.

Bien kan p.gr. af træthed beklageligvis ikke flyve og den må økonomisere meget med sine sidste kræfter - altså tænker den sig grundigt om - og efter en lille tid finder bien en løsning, som den mener giver den korteste vej.

- Bien når frem, og overlever!

Hvilken vej valgte bien?



4 x 4 x 10m lang

Blandt rigtige løsninger, som når frem til redaktionen senest 12 marts 95, deltager 3 i lodtrækning om hver en diskette med indhold fra MCUG's 'software-lager'. Eventuelle ønsker kan anføres sammen med løsningen.

LINUX bøger: De mest populære kapitler fra DR$_X$.LINUX bog. CD versionen inkluderer SlackWare-sættet af LINUX operativsystem og Utils. på disken
LINUX Getting Started-Documentation,
LINUX Getting Started-Doc. w/SlackWare, w. CD

Strobel, S.	LINUX - vom PC zur Work-Station, (Grundlæggende)
Hetze, Sebastian,	LINUX AnwenderHandbuch und Leitfaden
Kirch, Olaf	LINUX Network
Strobel, Stefan	LINUX-Power Pack

Alle ovenstående LINUX bøger er fra GAD, Nørreport,

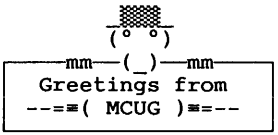
Linux - Forlydende: WordPerfect kan fås til LINUX i beta-udgave. (undersøges)

Nye filer kan nu hentes på BBS'et. MCUG AREA:MUG (fidonet/point/noder m.v.) +++

FAQPNT04.ARJ	20437	4-08-94	FAQ fra POINT_R23.PNT version 0.40
FIDOHIS1.ARJ	8379	23-02-92	Historien om FIDONET, fra 8. februar 1985
FIDOHIS2.ARJ	6798	23-02-92	Revideret historie om FidoNet, fra 20 august 1985
POINT.ARJ	7420	26-02-92	Point - Hvad er det? Af Martin Jønch(ex-2:231/36)
POINTINF.ARJ	2033	8-02-93	Information om point. Af Derly Lytken(2:234/71)

---

pluk fra SlackWare LINUX info..:
. Programming and Development Tools.
. Languages: GNU C, GNU C++, and Ada compilers. Assembler.
. Basic and Lisp interpreters. Fortran-to-C and Pascal-to-C translators.
. Debugging tools for X Window System and regular ASCII screens.
. Function libraries for development in all languages.



Misspelled? Impossible. My modem is error correcting.

"Spock, I thought you were dead!" "I rebooted, Captain."

"I'm not worth a thing before coffee break!"

All warranties expire upon payment of invoice.

Money can't buy love, but it can rent it for a while

Spare Wasser - bade mit einer Nachbarin

Coffee allocation error, . . . Operator paused . . .

This copy of ProBoard has been unregistered for 76 days!

Booting Warp II.....wuschhhhh.....

Come and feel Electric Dreams

Yell.....the Sysop is not here ...try again !

... three beeps means trouble's coming

If at first you don't succeed, put it out for beta test.

Aus Spass wurde Ernst, Ernst ist jetzt 3 Jahre alt.

Itsi Bitsi Teeny Weeny Honolulu Strandbikini.

...for more information, press the ANY key !

My best view from a Window was through OS/2

IBM = International Bubblegum Machines

Nothing smells stronger than Al's socks..

Hasta pronto, Kai Stukenbrock

Smile ... tomorrow will be worse !

Who is General Failure and why is he reading my disk?

If people behaved like politicians, you'd call the cops...!

Tagline Lotto: ▓▓▓▓▓▓▓▓▓▓<- Scratch here to reveal prize

8ung !!! Alles mitgem8 !!!

---------- der er flere! :-))

Linux.announce Tue Dec 20 17:08:31 1994

DR LINUX - The Complete Linux Documentation Book.

DR-LINUX.fd2

Dr Linux - or Doctor Linux - as in "Just What the Doctor Ordered" for managing your Linux installation. It is the perfect How-to and reference manual for all Linux distributions and CD's* including Slackware, Yggdrasil, Redhat, Tamu, and for CD packages from Infomagic, Walnut Creek, Morse (Slackware Professional), Trans-ameritech, and all the others.

This is the second edition of the Linux Documentation Project book published by LSL, this time in conjunction with ACC Corp (publishers of the "PC Unix and Linux catalog") and Lasermoon of Plymouth, UK.

This **1,180** page volume is the complete works of the Linux Documentation Project in a convenient and attractive printed book. It contains all the most recent editions of the documents listed below with many new documents as indicated:

- "Linux Installation and Getting Started" 2.1.1 by Matt Welsh.
- "Kernel Hackers' Guide" 1.0 by Michael K. Johnson
- "Networking Administrator's Guide" 1.0 by Olaf Kirch

[NEW!] ▪ "The Linux Users' Guide" 0.4 by Larry Greenfield
[NEW!] ▪ Alpha release of "System Administrator's Guide" by Lars Wirzenius
[NEW!] ▪ LILO Users Guide
[NEW!] ▪ LILO Technical Manual

- "How to" Guides on the following subjects:

[NEW!] Bus Mice	[NEW!] CDROM
Dos Emulation	Distributions
Ethernet	Floppy Tape
Hardware	Installation
[NEW!] Japanese Extensions	[NEW!] Keyboard Mapping
MGR Graphics	Mail
Networking	[NEW!] Network Information Systems (NIS)
Usenet News	[NEW!] PCI Bus
Printing	SCSI
Serial	Sound
Term	Tips
UUCP	[NEW!] Wine (Frequently Asked Questions list)
XFree86	

AKA: "The Bible"
This is the second edition of this book and is co-published by Yggdrasil Computing under a different cover (same contents) with the name "The Linux Bible".

DR LINUX SUPPORTS LINUX DOCUMENTATION PROJECT
$1 of all sales is donated to The Linux Documentation Project (LDP). The LDP is the group of talented volunteers who have written these documents and for whose work we are very grateful. This donation is the same, regardless of whether you buy Dr Linux directly from us or through a reseller, and is also being done by Yggdrasil for their version.

TELL YOUR LOCAL BOOKSTORE!
An incredible value to end users, this is also a very profitable addition to it for you - at the incredible reseller discounts of up to 65% off the list price - they will thank you!

RESELLERS WANTED!
This book has been attractively designed for retail display with great margins and is selling like hotcakes to the rapidly expanding community of Linux users at academic, corporate, research, and home sites around the world. Please call for quantity and reseller discount schedules.

HOW TO ORDER:
ACC Bookstores, "Home of the PC UNIX - Linux Catalog".
1 (800) 546-7274
info@acc-corp.com

Henvendelse: Frank Damgaard

```
------------------------------------------------------------
DR LINUX - The Complete 1,180 page Linux Documentation Book.
------------------------------------------------------------
List Price:  $49.95
```

Dr Linux is available from many resellers at a wide variety of prices.
As an example the ACC Bookstore Catalog sells Dr Linux for: $37.95

Please contact us by fax or phone for dealer, reseller, and quantity discount pricing.

```
Single Copy Shipping:
------------------------------------------------------------
USA:          UPS Standard: $ 5.95     Overnight:  $11.95
International: Airmail:     $19.95     Express:    $39.95

We accept:
-Major Credit Cards,  -Personal and Corporate checks, -Corporate
 and Academic purchase orders (subject to credit approval)

Orders can be phoned, faxed, emailed, or mailed to:
ACC Corp.  136 Riverside Ave, Westport CT  06880-4606
Tel:       (800) 546-7274  or  +1 (203) 454-5500
Fax:       (203) 454-2582
Email:  orders@acc-corp.com
ACC Bookstores
"Home of the PC UNIX - Linux Catalog"
1 (800) 546-7274
info@acc-corp.com
```

--
An serious typo was included in the recent announcement about the Dr Linux - The Complete Linux Documentation Book.
The version of Michael K. Johnston's "Kernel Hacker's Guide" included in Dr Linux is version 0.5, This is the same version of the "Kernel Hacker's Guide" (version 0.5) that is included in Yggdrasil's "Linux Bible" and other versions of this book.
(Not version 1.0 as stated in posting). We regret any confusion caused by this error.

Cheers, Bob. ACC Bookstores "Home of the PC UNIX - Linux Catalog"

Ny fremstillingsmåde for printplader

PCB-TONR.195

PCB med Direkte Toner metode.

Den hidtidige procedure med at overføre et pcb-layout fra fx papir til en kobberlaminatplade er enten dyr (fotografisk) eller omstændelig (silketryk mv.) ikke mindst, hvis det er til hjemmebrug.

En udbredt måde for fremstilling af trykte kredsløb er foto-metoden. På en printplade, som er forsynet med en pålagt / påsprøjtet lysfølsom emulsion, lægges en klar folie/film, på hvilken det ønskede ledningsmønster er anbragt. Efter belysning i passende tid med en UV-lampe kan pladen lægges i fx en Natriumhydroxid-opløsning for fremkaldning. Dvs. på den del af kobberet, som ikke ønskes, fjernes emulsionen og i næste trin kan det frilagte kobber fjernes i et ætsebad.

Karakteristisk for den skildrede fremgangsmåde er brugen af et lysfølsomt lag / foto-emulsion. FotoLak (vædske) kan fx sprøjtes eller slynges på, - FotoResist, en tynd film kacheres (at fastgøre, pålægge) på printpladen.

I hvert fald er der en del arbejdsgange at gennemgå: et foto-lag, pålægning, belysning, fremkaldelse, evt. fixering, skylle med vand og tørring - inden man når til det færdige resultat.

Denne vej er omstændelig, fuld af faldgruber, og for hvem, der er mindre øvet i denne metier, ikke let at gennemføre med succes. Dertil kommer at fotometoden er dyr, når man tænker på at en fotopositiv belagt printplade fra en bestemt størrelse koster omkring 3 gange så meget som en ubehandlet.

Direkte Toner Metode

Bag denne metode gemmer sig i det væsentlige at den for fotokopimaskiner og Laserprintere anvendte toner er en glimrende foto-resist.

Det er ikke mindst fortræffeligt, fordi Toner består af kemisk helt trægt kulpulver og af et bindemiddel (Kunst-harpiks eller Polyethylen) som ikke angribes af syre.

En yderligere vigtig ting er, at de fleste Laserprintere (med en opløsning på mere end 300 dpi) kan levere print-layouts i målestok 1:1 med en acceptabel kvalitet.

Med software kan man - når det ønskes - omstille til 2:1 eller 4:1 formater, positiv eller negativ og uden nogen form for fotografisk arbejde.

Resten af fidusen / trick'et består nu i, på passende måde, at overføre toner-masken til printpladen. Det kan ske ved at anvende en indtil 150 °C. varmestabil, transparent, kunststof-folie TEC 200,.

Folien er behandlet kemisk, så den kan modtage toneren, men også afgive denne 100%.

Folien er altså i princippet kun et indskudt transportmiddel for den af toneren bestående maske. Fremstillingen til færdigt print forenkles som følger:

- Tryk (LaserPrinter) eller kopiering (alm. kopimaskine) på folie i målestok 1:1.
- Overføring af toner-mønsteret, under tryk og varme, fra folien til kobberlaget på printpladen.
- Ætsning af printpladen på sædvanlig vis.

Resten forklares her nærmere.

1. Skridt: Kopiering

Det dybsorte print layout kopieres over på folien TEC 200 (A-4format). Folien lægger man - over/underside er ligegyldigt - i kopimaskinens papirmagasin, Sværtningsgraden skal være indstillet til høj, men dog ikke så høj at lederbanerne flyder sammen.

Først laves en papirkopi til prøve. Det er ikke nødvendigt at tonerkopien på folien fremtræder dybsort, da det til kulpulveret anvendte bindemiddel egentlig i sig selv er nok til at beskytte mod ætsemidlet, men det kan man ikke se lige så tydeligt.

Skulle layout'et ikke have været spejlvendt, så laver man en kopi af det på folien kopierede lederbane-billede, hvorefter man

så kan lægge denne kopi spejlvendt i kopimaskinen.

TEC 200 folien tåler temperaturer indtil 160 °C. uden nævneværdig skrumpning. Dimensionsforvrængning af layout'et, som kan vanskeliggøre isætning af større komponenter, som fx IC'ere eller kontaktlister, synes ikke at forekome, forudsat at man benytter en moderne, rimelig nøjagtig kopimaskine.

2. Skridt: Overførsel til printpladen

Efter at have kopieret layout'et til folien skæres dette fra A4-formatet.

Foliestykket lægger man, med printmønsteret nedad, på printpladens kobberside og fæstner folien med et stykke klæbebånd. Selve overførelsen laves sådan:

Opvarm printpladen til ca. 140 °C. på en varme- eller kogeplade evt. på et omvendt strygejern. Overfør v.hjælp af en gummirulle (fx fra fotoudstyr) med et let tryk på printpladen. Afkøl printpladen til stuetemperatur og fjern forsigtigt folien. Selv tynde lederbaner kan overføres fint.

Det har vist sig praktisk, at opbevare printplader med påført folie i køleskab og så trække den let-klæbende folie af når pladerne skal bruges. I nogle tilfælde kan det være en fordel at lægge printpladen med folie i ætsebadet hvorved folien selv løsner sig.

For lige at bemærke: Det på folien overførte mønster behøver ikke at være lystæt, idet lederbanerne (toneren) ved overførslen til kobberet smeltes fast og danner en lak-lignende, syrefast hinde.

Efter varme-kachering af lederbanerne (toner) kan folien langsomt og forsigtigt trækkes af. Herefter er pladen klar til ætsning.

Selvom toner-materialet til en vis grad er gnide- og ridsefast, bør man lade folien sidde på som ekstra beskyttelse, hvis man ikke skal bruge printet straks.

3: Skridt: Ætsning

Printpladen kan ætses straks efter overførsel af mønsteret. Til ætsning kan bruges alm. anvendte midler, og efter ætsningen skylles pladen med rigeligt vand.

Toner-rester kan fjernes med fx acetone, fortynder eller lign. opløsningsmiddel.

Den her beskrevne fremgangsmåde til fremstilling af en printplade er i mange situationer betydelig nemmere og hurtigere end ved adskillige andre metoder.

Oversat (uden ansvar - *red.*), fra *Elektor 11/1994.*
Folier kan leveres af nedenfor nævnte firma,
evt kan andre materialer med tilsv. specifikationer bruges.

```
Chemitec GmbH. / Auf der Winneburg 18 / D-56-814  Ernst
Telf.   +49 -2671 1631,      Fax. +49 -2671 3284

10 stk folie A4 incl. forsend. 19,50 DM (minimum 10 stk)
 1  -  Gummirulle               20,50 DM
```

ADRESSER, SOFTWARE & DISKETTER

Husk, ved diskette-bestilling, at oplyse om diskformat!

Volume fra bibliotek (3.5"/5.25") incl. disk & forsendelse 20,- kr.

Bestyrelsen:

Formand:
Frank Damgaard
Kastebjergvej 26A
2750 Ballerup
4497·3747

Kasserer:
Lars Gråbæk
Esbern Snaresgade 6
1725 København V.
3123 9236

Anders Otte
Grønnevej 261, 13
2830 Virum
4285·1645

John B. Jacobsen
Lyshøj Allé 20, 3th.
2500 Valby
3116·1393
e-mail:
johbjbj@inet.uni-c.dk

Thomas Jørgensen
Sct.Annægade 53-1h
1416 København K.
3154·7868

Donald Axel
Saxenkolvej 26 A
3210 Vejby
4870 6913

Vagn Nielsen
Klintevej 33
2700 Brønshøj
3128·2154

Viggo Jørgensen
Fensmarks Allé 6
3520 Farum

Bibliotek
Giro 5·68·6512

Frank Damgaard
Kastebjergvej 26A
2750 Ballerup
4497·3747
(man-tor 1730-1830)
e-mail: frank@diku.dk

Bulletin Board:

Telf. **3160·5319**
Åbent hele døgnet
300 - 14400 bits/sec
V32bis, V42bis, MNP5
8bit 1 stop, ej paritet
SysOp: Vagn Nielsen

Redaktør: Viggo Jørgensen, FensmarksAlle 6, 3520 Farum, 42 95 32 01

MCUG MicroComputer-User-Group

...en ikke-kommerciel forening for brugere af mikro-datamater, vore biblioteker understøtter IBM-PC og dermed kompatible mikro-datamater, samt CP/M.

Foreningen drives på frivillig basis og er rettet mod dem, der ønsker at få mere ud af deres computer end blot muligheden for at køre standard programmer.

Foreningen søger at støtte medlemmerne i brugen af deres computer ved arrangement af:

1. Medlemsmøder, hvor man kan mødes og snakke sammen, udveksle ideer, hente inspiration og få hjælp med problemer vedr. computere.

2. Fællesindkøb, hvorved vi kan opnå rabatter på komponenter, tidsskrifter, bøger, software, hardware etc.

3. Foredrag hvor folk, der ved mere end gennemsnittet om et emne, kommer og fortæller, så vi alle kan få udbytte af det.

4. Udsendelse af et aperiodisk nyhedsbrev, som udkommer på diskette i standard IBM format, med nyheder, tips, anmeldelser af bøger, soft- og hardware, kataloger fra foreningens software bibliotek samt diverse programmmer / shareware programmer.

Udgivelse af medlemsblad/hefte (almindeligvis 4-6 gange årligt.) med stof af forskellig art. Her kan medl. bringe artikler, små-nyt, spørgsmål, gratis (private) annoncer, osv.

Et bulletin board er til rådighed for medlemmerne, således at disse via modems kan udveksle meddelelser, programmer og få informationer, der stadig er o*vnvarme.*

Foreningen hjemtager public domain/shareware og mod en lille kopiafgift stiller dette til rådighed for foreningens medlemmer. Kopiafgift (pt. 20 kr./volume) skal dække omkostninger og distribution samt udgøre grundlag for biblioteks-udbygning.

Public domain programmer er progr., der som navnet siger, ikke er omfattet af copyright og derfor kan distribueres frit. Det omfatter bl.a. programmeringssprog, tekstbehandling, regneark, database-programmer - endv. mange spil og værktøjer for blot at nævne et udsnit.

Kontingentet er 225 kr. årligt og gælder 1 år fra indmeldelsesdatoen.

Indmeldelse kan ske ved indbetaling af kontingentet (225 kr.) på girokonto:

5 68 6512 MCUG Denmark, Esbern Snaresgade 6, 1725 København V.

Yderligere oplysning kan fås hos formanden eller kassereren på telf.:

4497.3747 & 3123.9236 samt BBS 3160.5319