

DKUUG NYT | 149 | MAJ 2005

VEJEN TIL VIDEN OM ÅBNE SYSTEMER OG INTERNET

Free Beers and Naked Women s2 > Rapportage fra Usenix 04 s6

Free Standards Group og Linux Standards Base s4

Bestyrelsens beretning 2003/2004 s16

DKUUG *nyt*



Danmarks

største

it-nyhedsbrev ...

... og naturligvis det bedste og flotteste!

42.304 it-professionelle modtager IT-avisen ComONs
nyhedsbrev 2 gange dagligt.

Vil du også opdateres med de seneste it-nyheder?



Gå ind på www.comon.dk og bliv en del af succesen!

ncom

GNU'er i Tanzania

AF OLE TANGE | Foto: Hans Schou

Mine forældre ejer et stykke land i Tanzania. Tange's Field hedder det. Det er, så vidt jeg ved, en mark med nogle træer, som de lokale vedligeholder; for jeg har aldrig selv været nede og se marken. Det har mine forældre heller ikke, men min far er pludselig blevet interesseret i at tage derned.

Interessen kom for en måneds tid siden, hvor han hørte om et landsprojekt, hvor en frivillig tog forældede computere med til et uland og stillede dem op der. Projektet fik sponsoreret software til computerne af Microsoft – ganske gratis. Eller også var det ikke helt gratis. Som min far udtrykte det, så ville de næppe have betalt for softwaren alligevel, så Microsoft ville ikke have tjent på dem; men nu hvor modtagerne er identificerede, så kunne Microsoft muligvis kræve penge for opdateringer. »First shot free.«

Min far blev temmelig harm over projektets kortsynethed: Hvordan skulle de kunne få råd til fremtidige versioner? De havde jo heller ikke selv mulighed for at udbedre software, hvis den viste sig at have sikkerhedsproblemer. Og hvad hvis Microsoft pludselig holdt op om med at oversætte programmerne til swahili? De ville heller ikke (lovligt) kunne give en kopi til nabolandsbyen, hvis nabolandsbyen købte en gammel computer. Alt i alt syntes han, der var ret mange problemer ved at benytte Microsoft.

Næh, min far ville have, at de skulle benytte fri software. For med det ville de kunne få fremtidige versioner billigt, og hvis de havde kompetencen, kunne de selv forbedre softwaren – både sikkerhedsmæssigt og med oversættelse til swahili. Og de må endda forære en kopi til nabobyen.

Det var så her jeg blev involveret i processen. Jeg anbefalede ham, at prøve en Knoppix-CD, og hvis han fandt den brugbar, så måtte vi få den oversat til swahili. Vores gæt er, at Tanzanianerne næppe de første mange år får brug for mere software end der kan være på en enkelt CD. Det er muligt, at de kan få brug for andet software end der er på en normal Knoppix-CD, men da der alligevel skal laves noget tilpasning af CD'en (nemlig sproget), så vil det ikke være det store problem at installere lidt andet software.

Så nu prøver min far at få skabt de rigtige kontakter. Han forventer, at første rejse til Tanzania sker inden for et år, og at hans rolle bliver at lære beboerne at bruge softwaren. Min rolle bliver at få maskinerne til at fungere.

Så for mig kommer fri software muligvis til at betyde, at jeg rejser til Tanzania. ■



Free Beers and Naked Women

Rapportage fra Usenix 04 i Boston, USA
ved Morten S. Olsen og Kenneth Geisshirt, Medical Insight A/S,
Hovedgaden 451C, 2640 Hedehuse

Free Beers

Sjældent har vi drukket så mange gratis øl som den uge, hvor vi taget til Usenix' årlige tekniske konference. I år var den henlagt til Boston, så vi mødte ved flere lejligheder Samuel Adams.

Som noget nyt havde programkomiteen besluttet at i stedet for en stor reception, var der happy hour mellem 17 og 18 ##hver dag##. Og det elegante var, at der var en sponsor til hver dag. Præcis mellem 17 og 18 var det ikke andet i programmet. Vi begyndte hver dag kl. 9 og efter kl. 18 var det et mere uformel program i form af BoFs.

Plenarforedrag

Programkomiteen havde virkelig forsøgt at gentænke en teknisk konference. Så i stedet for de stort anlagte »keynote speakers«, begyndte hver dag med et plenarforedrag. Hver af de fem foredragsholdere kunne uden problemer have fungeret som »keynote speaker«. De fem plenarforedrag var:

- Eric Allman om spam
- Rob Pike om Googles serverfarme
- Alan Nugent om Novells syn på Open Source
- Bruce Scheier om sikkerhed
- Eliot Lear om netværkskompleksitet

Det er godt at begynde dagen samlet – og vågne op med lidt kaffe og et teknisk interessant foredrag.

Den første dag blev der udover plenarforedraget, også uddelt både »STUG« - og »Flame«-prisen. Begge priser blev givet til Doug McIlroy for hans arbejde med at udvikle et af de mest grundlæggende koncepter i Unix, nemlig »pipes«, og nogle af de mest basale utilities der udnytter dem, deriblandt diff(1) og sort(1).

Guru-sessioner

En række guruer har blevet inviteret til på halvanden time at give en indføring i deres specialer.

Steve Rees fra IBM/Canada fortalte om hvordan DB2 under Linux kan komme til at køre bedst muligt. En

deltager nævnte at han i den senere tid har målt at Linux har 2-10 gange bedre price/performance i forhold til Solaris. Den sædvanlige diskussion om Open Source er seriøst eller ej brød ud – med Steve Rees på Open Source-siden!

Ydeevnen af NFS har altid været et stort samtaleemne, men ved Usenix 04 var gurun ingen ringere end Tom Talpay. Tom havde mange gode tips til hvordan NFS kan optimeres på flere forskellige styresystemer. NFSv4 er på trapperne, og referenceimplementationen sker på Linux.

Det er jo ingen hemmelighed at den ene af os interesserer sig for Linux-klynger. Derfor var det også naturligt for Kenneth at deltage i to guru-sessioner om klynger. Den ene var om det distribuerede filsystem Lustre, hvor Phil Schwan (en af arkitekterne) gennemgik de tekniske aspekter af filsystemet. Blandt deltagerne var der flere brugere, og det var interessant at høre deres konkrete problemer.

Den anden klynge-guru var Greg Bruno som er hovedudvikler af Linux-distributionen Rocks Clusters. Det er en distribution som er målrettet mod Beowulf-klynger, dvs. klynger som skal bruges til beregninger.

På programmet var lovet en X11 guru-session med Keith Packard, som Morten havde set frem til. Han var desværre forhindret i at komme, omend han dukkede op senere på freedesktop.org BoF'en. I stedet var Bart Massey blevet udnævnt til X guru, og der blev snakket en del omkring den nuværende status med XFree86 og X.org. Han havde et par af sine studerende med; Eric Anholt som havde lavet en ATI Radeon driver med acceleration af XRENDER, og Jamey Sharp som arbejder på XCB, et ambitiøst projekt der har som mål at erstatte selve C grænsefladen til X protokollen, »Xlib«, med et moderne og mere strømlinet bibliotek.

Paper track

Et af de første tekniske præsentationer var også en af de bedste. En kerneudvikler fra Sun der, skal det siges, ikke stod tilbage for Kim Schumacher i hur-

tignak, fortalte om DTrace, der kan bruges til at instrumentere Solaris-kernen. Lidt i stil med Linux Trace Toolkit, dog blot meget mere avanceret. Som demonstration præsenterede de en ren detektiv-historie, hvor et performance-problem på en server der kørte tusindvis af tynde SunRay-klienter efter meget tracing blev sporet til blot 6 brugere der alle kørte en enkelt X applikation der opførte sig meget asocialt. Applikationen var en lille applet til GNOME som tjekker aktiekurser.

Freenix-nyt

Indenfor Open Source/Free Software (som generelt kaldes Freenix i Usenix) sker der rigtig meget. Kirk McKusick havde inviteret alle BSD-varianterne (også den nye DragonFly) til at give en statusrapport. Desværre kunne Apple-folkene ikke komme, og OpenBSD-folkene kunne ikke blive enige om hvem de skulle sende. Men FreeBSD og NetBSD gav gode opdateringer af deres respektive systemer. Glædeligt er det at høre at FreeBSD næsten er kommet af med den store kernelås. Den nye BSD-variant DragonFly introducerede sig selv i familien.

Linux var repræsenteret meget stort. En af de mere interessante foredrag var om et projekt hos AT&T hvor en masse gammel MVS-kode blev flyttet til en Linux-klynge. Koden er skrevet i Cobol og i MVS' Job Control Language. Heldigvis findes der en fri Cobol-oversætter til Linux, og de features, som manglede, kunne forholdsvis enkelt skrives.

Det var også et Linux-releteret bidrag som blev kåret som det bedste, nemlig projektet Wayback. Wayback ligger sig oven på et alm. filsystem og alle filer er nu versionsstyret. Det betyder at alle ændringer gemmes og det er muligt at gå tilbage til en tidligere udgave. Projektet er forfatterens ph.d.-opgave, og det kan findes på <http://wayback.sourceforge.net/>.

De mange BoFs

Der var rigtig mange BoFs. Vi deltog i en del, f.eks. det store Linux User Group møde, hvor primært Linux-brugergrupperne fra Boston-området var repræsenteret. Miquel de Icaza fra Novell (og GNOME) fortalte om Mono-projektet ved at give en demonstration hvor han skrev et webbrowser.

Kenneth deltog i et par mindre besøgte BoFs, bl.a. en om GNU awk. Der sker en del spændende ting

med awk, f.eks. findes er et patch som gør awk XML-orienteret i stedet for linjeorienteret som i dag. Samtidig sker der en del internt i awk, herunder en bytcode-fortolker og en debugger. Kenneth fik sig også en god snak med flere GNU-folk ved en BoF arrangeret af Free Software Foundation.

Klapper man af SCO på Usenix? Ja, det lyder ikke som en oplagt ting at gøre, men det skete på en måde på BoF'en omkring SCO-sagens juridiske aspekter. En udvikler med en falmet »Santa Cruz Operation« t-shirt fortalte om de forskellige SCO'er, og belærte os om at SCO (version 0 - firmaet har eksisteret i mange udgaver) faktisk engang for mange år siden ##var## et fornuftigt firma.

Derudover var det en interessant BoF, hvor vi kom rundt om mange aspekter om sagen. Den tilstedeværende jurist havde en forbløffende evne til, endnu før et spørgsmål var afsluttet, at have den relevante lovtekst på projektoren. Efter lettere pres fra Rob Kolstad, der styrede BoF'en, stillede Dennis Ritchie sig op og snakkede kort om »Ancient Unix«.

Game show

Google var sponsor ved et stort game show. Dan Klein og Rob Kolstad legede TV-værter i en nørdet udgave af quiz på paratviden. Dan og Rob ##er## morsomme og publikum grinede sig halvt ihjel. Før showet serverede Google øl (Google bar?) og uddelte merchandize – kuglepenne, t-shirts og blinkende logoer.

Boston by night

Boston er en lille by – det er muligt at gå rundt i bymidten og nå det hele på et par dage. Vi skulle jo naturligvis lige nå at opleve Quincy Market og gå på Freedom Trail. Ølbølgen som skyller ind for Danmark i disse år, har også ramt Boston. Vi fandt et par lækre microbreweries f.eks. Boston BeerWorks (<http://www.beerworks.net/>). Kenneth er til pale ale og stout, og han fik valuta for pengene.

Naked Women

Tja, naked women så vi ikke noget til, selvom vi håbede at Valery Henson fra LinuxChix.org ville arrangere noget. Forhåbentlig kommer vi tilbage næste år og leder videre. ■

Free Standards Group og Linux Standards Base

Ved den Schweiziske Unix Konference (SUCON 04) gav Theodore Tso, bedre kendt som Tytso, en introduktion til Linux Standards Base (LSB) projektet og den organisation, Free Standards Group, som arbejder med at finde en formulering af Unix-standarden, sådan som den har udviklet sig. Formålet med LSB er at fremme et sæt af standarder, som vil forøge kompatibiliteten mellem Linux distributioner og gøre det muligt for applikationer at køre på et LSB-compliant system. Desuden vil LSB ved at koordinere anstrengelserne gøre det muligt at få flere leverandører til at portere og udvikle flere produkter for Linux.

Det lange sigt er, at LSB, ved at sikre så bred tilslutning som muligt fra alle hold, vil kunne fremme den enkelte distributørs muligheder for at henvende sig til et stort marked - oftest en specialiseret niche, men en niche på et marked, som har taget Linux til sig.

Theodore Tso er en kendt figur i Unix verdenen, idet han er manden bag ext2 filsystemet, Linux extended file system version 2. Og i øvrigt er han også manden bag version 3, ext3, som er et journaliseret filsystem med en performance på linie med IBM's JFS.

I oplægget til foredraget fortælles det, at LSB standardiserer basis for Linux systemet og derved tillader ultimativ portabilitet af applikationer på tværs af systemer, som overholder standarden. Ikke desto mindre er LSB meget mere end en nedskrevet specifikation. Den indeholder test-suiter, en modelimplementation, så man kan se, at det faktisk virker, et build miljø, og en guide for udviklere af både Linux operativsystem - distributioner og Linux baserede applikationer.

Tytso's foredrag er forevigt i en .avi film, som uden videre kan afvikles med en nyere version af mplayer. Herfra har vi hentet nogle særligt prægnante bemærkninger. Filmen (filen) kan hentes her på http://mirror.switch.ch/sucon-04/theodore_tso-lsb.avi

For at demonstrere, hvilke ærgrelser, en Linux bruger kan komme ud for, nævner Tytso at der endnu ikke har været en release af C++ std.libraries, som var kompatibel med den foregående. Hans mening om C++ er provokerende:

[...] the C++ people haven't been able to get their act together, and every single C++ release has had some kind of incompatible change to it, at an API level, which means upgrade your C++ compiler, we compile all your libraries and programs, somehow this is considered acceptable, I don't understand, but, I guess that if you have bad enough taste to use C++ you are willing to live with that, I am not sure, but: This is a problem!

Bortset fra denne og lignende skrappe »opinionated« sætninger, så er Tytso's tankegang klar og præcis, hvilket er meget rart at konstatere, når man har betroet sine data til hans ext2 filsystem gennem godt 8 år nu!

Klarheden ses tydeligt i argumentationen for, hvorfor man skal tilslutte sig LSB. Der er to væsentlige grunde, den ene er, at US Defense kræver LSB compliance, en meddelelse som fik flere leverandører til at hoppe frem i rampelyset og erklære, at de altid havde elsket LSB - og den anden væsentlige grund er så, at LSB giver et større marked og mindre vedligeholdelse. LSB er udformet på en måde, så den enkelte leverandør stadig kan vedli-



geholde sin egen distribution med de kendetegn og de særlige fordele, som distributionen rummer.

Hver eneste beslutning i LSB er politisk, snarere end teknisk, men alligevel er det en »udvikler specifikation«. Tytso giver et eksempel: sighthandler data strukturen er i LSB specificeret helt præcist; ikke som i Posix, der siger, at den minimum skal indeholde sådan og sådan og så i øvrigt kan gøre som den vil, nej, i LSB skal den indeholde disse felter i denne rækkefølge, værsgod, færdig!

Derved bliver det muligt at bruge headersystemet til at teste for compliance med library, og det bliver muligt for udviklerne at fange fejl, fx. hvis en ændring i en library-definition vil brække applikationer. I libtiff havde en ændring af en integervariabel fra int til short bevirket en sådan fejl, uden at programmørerne havde været klar over det.

LSB bygger selvfølgelig på Posix, SVID, susv3 (Single Unix Specification version 3), ELF formatet, m.v.

Den måske nok vigtigste ting ved LSB er, at man sigter mod modularitet. Det skyldes flere forhold. For det første vil man kunne sige til de engagerede programmører indenfor et bestemt område, at de selv kan vedligeholde standarden. For det andet vil man kunne definere forskellige profiler, en for embedded systemer, en for GUI systemer og så fremdeles, uden at man derfor behøver at have helt forskellig basesystem. Ved at følge denne strategi håber Free Standards Group at kunne få flere virksomheder, brugere såvel som distributører og udviklere, til at tilslutte sig LSB.

Tytso har programmeret C og Unix siden 1987, og er i dag ansat som senior teknisk stabsmedlem i IBM Linux Technology Center.

Den nye LSB 2.01 er sendt til ISO for godkendelse. Dermed vil LSB effektivt være en overbygning – eller afløser, om man vil – for Posix standarden.

Vi vil følge op på artikelen med information på DKUUGs website. ■

**DKUUG's
bogholderi træffes
tirsdage mellem
11 og 16 på
tlf. 39 17 98 21**

Creating virtual »soft« devices with User-mode Linux

Sapan Bhatia & Laurent Réveillère

LaBRI/INRIA
ENSEIRB

1 Ave. du Dr. Albert Schweitzer
Domaine Universitaire
33400 Talence, FRANCE
{bhatia,reveillere}@labri.fr

Abstract

Developing device drivers can be highly tedious as it entails direct interaction with hardware devices, which are difficult to analyse in trying to find the cause of unexpected behaviour. User-mode Linux (UML) [3, 4] simplifies this task by allowing developers to test and debug their device drivers in user-space. In this paper, we describe a systematic approach to create virtual *soft* devices using UML for the purpose of testing device drivers while they are still in the stages of development. Soft devices run as user-space processes and can have a GUI interface. We have used the existing emulation capabilities of UML and extended them by adding some of our own. We have designed a language named *Saint* to specify soft devices, and implemented a virtual coffee-machine soft device as a proof of concept.

1 Introduction

Device drivers are a key component of operating systems. They insulate application developers from the details of underlying hardware, thus implementing the mechanism of the software interface with the device. The use of this mechanism, or the actual policies are left to application developers. Writing device drivers is a highly rigorous task especially when considering *hardware operating code* (i.e. code interacting with hardware). This layer of code is low level and known to be *error prone*. This is aggravated by the fact that minor errors can be expected to have disastrous effects on the system, as hardware devices are not robust to erroneous code.

It is thus imperative that device drivers be thoroughly tested in a mock test environment before they are actually made to interact with real hardware. User-mode Linux (UML) [3, 4] provides such an environment by running the Linux kernel in user space. This means that device drivers are executed in user con-

text, reducing the penalty of faulty code. Instead of system crashes, faulty device drivers cause exceptions in the UML kernel, which causes the UML processes to terminate. Device driver developers can also make use of user-space libraries and applications for extensive logging and debugging.

UML can also be made to emulate I/O memory and interrupts. Files on the host system can be mapped into UML as I/O memory. Similarly, SIGIO signals can be used to generate interrupts inside UML. This suggests that one should be able to emulate a complete device, and use the emulated device to test and experiment with a device driver. One of the things that UML lacked in this respect was the ability to emulate I/O port memory. We have implemented I/O port emulation for UML, which is to be merged into the UML tree in the mainline Linux kernel. Apart from this, we modified the timer modules for more precise timing, as device drivers often rely on it, and the interrupts sub-system to be able to pass a file to UML so that device drivers could transparently register IRQ handlers. This file would be used to generate interrupts inside UML.

In this paper, we show how one can use these emulation capabilities of UML to create virtual soft devices to be used to test device drivers. Such devices run as user-space processes and can have GUI front ends to display register states and give information about the operation of the device interactively. We also envisage this concept to be used to profile actual devices. A software implementation of a real hardware device can be used to analyse possible settings and configurations by subjecting the device to real application loads. I.e. processes in UML could use the device oblivious of the fact that it is emulated, and the process emulating the device could gather valuable information from the work load. To simplify the creation of soft devices, we have designed a language named *Saint* to specify them. *Saint* is a *Domain Specific Language (DSL)* [?] that generates stubs used in the device implementation.

To illustrate the concept of soft devices, we have implemented a conceptual device, a coffee machine, as a soft device and written a device driver to control it. The coffee machine application runs on the host machine and has a GUI interface. Within UML, the coffee machine can be manipulated using port memory and hardware interrupts. This device has already been used in a senior year device drivers course.

In Section 2, we give a brief overview of User-mode Linux. In Section 3, we outline how we have emulated hardware resources to create soft devices. In Section 4, we discuss the possible uses of soft devices. In Section 5, we describe our coffee machine device, which we use as an example while outlining the procedure to create and use soft devices. In Section 6, we detail the process of creating and using a soft device and in Section 7 we discuss how device drivers access it from UML. Finally, in Section 8, we describe Saint, a language we have designed to specify soft devices and simplify the process of their development.

2 User-mode Linux

User-mode Linux is a port of the Linux kernel that runs on Linux in a set of processes. The result is a user-space virtual machine using simulated hardware, constructed from services provided by a Linux kernel running on hardware (*a host kernel*). A Linux virtual machine is capable of running all of the applications and services available on the host architecture. Instead of a hardware architecture, as is usual, UML uses the system call interface of the Linux kernel as its underlying hardware interface. The code that implements this is under the arch interface of the Linux kernel. The rest of the kernel is an unmodified, full-fledged Linux kernel with all its features including scheduling, memory management and dynamically loaded modules.

UML, like the actual kernel, distinguishes between a privileged *kernel mode* in which only trusted code runs and a non-privileged user mode in which operations are arbitrated before being executed. This is done using the ptrace interface to intercept system calls. Thus, system calls issued in kernel context are executed directly without any arbitration, while system calls issued in user context are intercepted by the UML kernel, which annuls them on the host kernel and emulates them. Virtual memory is implemented by treating a file on the host system as a pool of physical memory. The virtual memory system in the Linux kernel then uses this pool of physical memory as usual. When a page needs to be mapped into virtual memory, a corresponding »physical page« in the file is mapped into an appropriate spot in the virtual memory in UML. UML implements I/O device interrupts with SIGIO, page faults using SIGSEGV and timer interrupts using SIGALRM.

3 Emulation of hardware resources

In this section, we discuss the emulation of hardware resources, which is used extensively in soft devices. Of the resources that are emulated, I/O memory emulation was fully implemented when we started this project, interrupt emulation needed modification and I/O port emulation needed to be implemented completely.

3.1 I/O memory emulation

I/O memory is the main mechanism used by device drivers to communicate with devices. UML has support for I/O memory emulation. This means that a host file can be specified as an I/O region in the UML kernel, so that drivers can locate it and use it as desired, just as they would use physical memory. If UML is executed with the argument

iomem=softiomem,/tmp/softiomem, then a driver can locate the corresponding memory area in UML using the `find_iomem` routine as follows.

```
device_memory = (void *) find_iomem(»softiomem«,
&iomem_size);
```

This is used by soft devices to make their device memory and memory mapped registers accessible to the device driver.

3.2 Interrupt Emulation

In UML, interrupts are emulated using signals and in particular, the SIGIO signal. Every file descriptor corresponds to a different IRQ. Interrupt handling in UML is a bit more tedious than on a host kernel. Other than requesting an IRQ and handling interrupts, drivers must also manage the files which will be used to deliver signals. They must also disable the corresponding file descriptor while the interrupt is being handled and re-enable it once it has been handled. This is undesirable when device drivers are being tested, as they are intended to run on the host kernel, and should run under UML as is. To make this possible, we made this process transparent by factoring out the code that handles files and enables and disables file descriptors. For the latter, we used the `irq_desc_t` interface, which is invoked at various stages of handling an interrupt.

3.3 I/O port emulation

We had to implement I/O port emulation since it did not exist in UML to begin with. Under Linux, processes gain access to I/O ports by using the `iopl` and `ioperm` system calls. Every time an I/O request is executed by a process without having done so, an exception is generated by the processor. This exception is relayed to the offending process as a `SIGSEGV` signal with information such as the fault address, the fault type etc.. We intercept such exceptions, nullify them and go on to interpret the faulting instruction i.e. the I/O call. We apply the result of the instruction to an area of memory we share with the soft device. The soft device is then sent a `SIGUSR1` signal to trigger associated side effects. Finally, we implemented the `iopl` and `ioperm` system calls to manage access permissions to ports in UML.

4 Usage scenarios for soft devices

Although so far, we have only used our work for teaching a device drivers course in which students were expected to write a device driver for a soft device, we envisage many diverse uses of soft devices:

- *Making device drivers more robust.* By implementing the behaviour of a device, fully or partially, one can test device drivers in a convenient way. The coffee

machine device we have implemented illustrates this. Furthermore, since processes are far more flexible than actual devices, they can be manipulated to create test cases for device drivers which would not be possible to create with real devices. Sometimes, unfavorable conditions may be created by implementing deviant or inconsistent behaviour in the soft device. Unified device drivers for a range of models of a particular device can also be tested in this way, by configuring the soft device to behave specific to particular models.

- *Teaching device drivers and operating system concepts.* Since soft devices are processes that can be restarted when they crash, and made to display an error condition instead of indicating it with a total system freeze, they are especially well suited to the purpose of teaching how to write device drivers.
- *Profiling hardware devices.* Since soft devices can be subject to processes run inside UML, they can be implemented in a way to collect critical information from real-world situations. This goes much beyond simulations, as it makes use of real-world work loads. In an extreme case, it might actually be possible to implement a device in circuit detail so as to record information pertaining to hardware modules and logic elements.

5 The coffee machine device

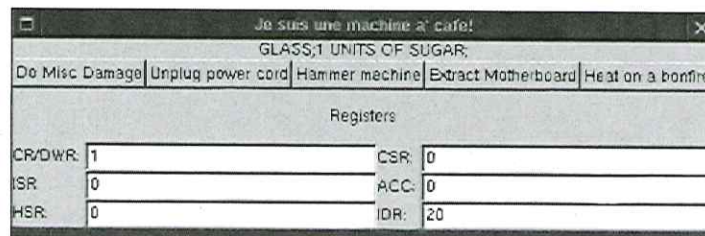


Figure 1: The coffee machine

We have implemented a conceptual device, a coffee machine as a soft device. On the host, the device is a process with a GUI interface (as show in figure 1. Ins-

ide UML, this device is accessed using five registers, which can be used to invoke different operations. These are illustrated in Table 1.

BAddress	RD	WD
0h	Data Waiting Register (DWR)	
1h	Interrupt Status Register (ISR)	Reserved
2h	Hardware Status Register (HSR)	Reserved
3h	Container Status Register (CSR)	Reserved
4h	Accumulator (ACC)	Reserved
5h	Reserved	Indexed Data Register (IDR)

Table 1: Direct Registers

As an example to show how the registers are used, the *command register* is described below.

7	6	5	4	3	2	1	0
AS2	AS1	AS0	PS1	PS0	IAD2	IAD1	IAD0

Bit	Symbol	Description
D2-D0	IAD0 - IAD2	Index Address: These three bits select which indirect register is to be accessed through the <i>Indexed Data Register (IDR)</i> .
D3 - D4	PS0 - PS1	Push Select: These two encoded bits control <i>Push</i> operations. PS1 PS0 0 0 No Operation 0 1 Push a glass 1 0 Push a stirrer 1 1 Push items
D5 - D7	AS0 - AS2	Action Select: These three encoded bits control actions other than a push. AS2 AS1 AS0 0 0 0 No Operation 0 0 1 Eject the glass 0 1 0 Reset the machine 0 1 1 State Request 1 0 0 Lock 1 0 1 Unlock 1 1 0 Data Reading Acknowledgment 1 1 1 Reserved

The coffee machine also generates interrupts on certain events, such as error conditions and during state queries, to indicate that new data is available. The type of the interrupt is specified by the *interrupt status register*:



7	6	5	4	3	2	1	0
DW	-	-	-	CNE	OWW	HWF	OPC

Bit	Symbol	Description
D7	DW	Data Waiting: This bit is set when a byte is available in the Accumulator register (ACC). This bit is low when no more bytes are available through the Accumulator register.
D6	Reserved	Reserved
D5	Reserved	Reserved
D4	Reserved	Reserved
D3	CNE	Container Empty: Indicates that at least one container is empty. The Container Status Register (CSR) records status of the various containers. This bit is cleared by the coffee machine itself.
D2	OWW	Overflow: Set when no more items can be accommodated in the order. I.e., the dispensing compartment is full.
D1	HWF	Hardware Failure: Indicates a hardware failure. The Hardware Status Register (HSR) allows one to identify the cause of the failure. This bit is cleared by the coffee machine itself.
D0	OPC	Operation Completed: This bit is set by the coffee machine when an operation is completed. This bit is cleared when an operation is triggered. Note that no operation must be triggered when this bit is high. Otherwise, the result is unpredictable. However, a hardware failure interrupt should normally be generated.

6 Creating and using soft devices

To create a soft device, we first need to map an area of memory which we are going to share with UML. This can be done with the `mmap` system call as follows:

```
device_resources.base_address =
(unsigned char *)mmap (NULL, PAGESIZE, PROT_READ | PROT_WRITE,
MAP_SHARED,
device_resources.fd, 0);
```

Where `device_resources.base_address` is the base address of the shared mapping and `device_resources.fd` is the file descriptor specifying the file to be passed to UML, so that it can be mapped inside UML. Once we have this memory area, we can assign

a layout for it, depending on the specification of the device we are implementing. For example, for the coffee machine, we lay out the registers starting from offset 0:

```
reg_cr = (union CR *) device_resources.base_address;
reg_isr = (union ISR *) ((unsigned int) (device_resources.base_address) + 1);
reg_dwr = (union DWR *) ((unsigned int) (device_resources.base_address) + 1);
reg_hsr = (union HSR *) ((unsigned int) (device_resources.base_address) + 2);
reg_csr = (union CSR *) ((unsigned int) (device_resources.base_address) + 3);
reg_acc = (__uint8_t *) ((unsigned int) (device_resources.base_address) + 4);
reg_idr = (union IDR *) ((unsigned int) (device_resources.base_address) + 5);
```


We then install a handler for SIGUSR1, so that when the state of the registers changes, we can perform the necessary action.

To be able to generate interrupts, we need to create an object which can be used to dispatch SIGIO signals to UML. In the coffee machine, and in Saint, we use UNIX domain sockets. We create a socket as follows:

```
sock = socket(PF_UNIX, SOCK_DGRAM, 0);
addr.sun_family = AF_UNIX;
snprintf(addr.sun_path, 5, »%5d«, getpid());
err = bind(sock, (struct sockaddr *) &addr, sizeof(addr));
```

We generate interrupts by writing to or reading from the socket:

```
char c='1';
err = sendto(fd, &c, 1, 0, (struct sockaddr *) &sun, sizeof(sun));
```

I/O memory can be implemented in soft devices in much the same way as I/O ports are, with the difference that there are no side effects associated with reading or writing to I/O memory, and the device must poll for changes.

7 Accessing soft devices from User-mode Linux

Device drivers for soft devices can now be written in UML just as they would be on the host system. Registers can be accessed using I/O instructions, for example:

```
void push_milk()
{
    reg_idr.bits.ID0=1;
    reg_cr.bits.PS=3;
    outb(reg_cr.byte, ADD_CR);
    down_interruptible(&op_sem);
}
```

Device interrupts can be intercepted by registering an interrupt handler, as shown below:

```
result = request_irq(CAFE_IRQ, cafe_interrupt_handler,
    SA_INTERRUPT, »cafe«, (void *) last_status);
```

User processes in UML can use the `ioperm` and `iopl` system calls to gain access to a range of ports, and invoke the I/O instructions, in and out to read and write them as usual.

8 The Saint language

Before starting on a new language, we had the option of using Devil [6, 7], a language used to write

device drivers. However, we decided that we required only a subset of Devil, and had some additional requirements for which we would require extensions. Thus, we designed the Saint language. The purpose of the Saint language is to be able to generate code for common tasks in the implementation of soft devices.

The Saint stub generator also produces code to perform consistency checks at run time depending



on specified properties. For example, if a port is defined as *read only*, the generated code will generate an error condition if it is written to.

```

device example {
  ports (0..80) {
    (0..4) read;
    (5) read,write;
    (6..79) read;
    (80) {
      (0..3) read;
      (4..7) write;
    }
  }

  mem (0..100) read,write;

  registers {
    ports {
      (0) ACC;
      (1) ISR;
      (2) {
        (0..1) STATUS;
        (2..7) ERR;
      }
    }
  }

  handlers {
    ports {
      (0..4) handle_first_four_ports(ACC,ISR,STATUS,ERR);
      (80) {
        (2..7) handle_error(ERR);
      }
    }

    faults handle_fault;
  }
}

int handle_first_four_ports(unsigned int &ACC, unsigned int &ISR,
                           unsigned int &STATUS, unsigned int &ERR) {
  /* fill in your code here */
}

```

Figure 8 shows an example device specification in Saint. The ports section defines 80 ports, addressed from 0 to 100, and specifies their properties with respect to whether they can be read or written to. The read, write and none properties can be specified for one or more ports, or one or more bits for a particular port. This scheme is followed in the definition

of I/O memory, registers and event handlers as well. For example, the second line in this section defines the port at offset 5 as readable and writable. The mem section follows the same semantics as the ports section, and is used to define I/O memory. In this case, it defines 100 memory locations addressed from 0 to 100, and that they can be read or written to.

The registers section defines named registers which will be passed to event handlers defined in the *handlers* section. Registers can span a range of ports, or a range of bits in one port. For example, the fifth line defines the register ERR to span 6 bits (2-7) of port 2. The handlers section defines a set of handlers for when ports are read and written to. The first line of this section specifies that the *handle_first_four_ports* function be called with the registers ACC, ISR, STATUS and ERR as arguments. The last line of this section specifies that the *handle_fault* function should be called in case of faults, such as writes to read-only ports or memory.

Figure 8 gives an example of generated stub code. The *handle_first_four_ports* function will be called when one of the first four ports is written to, after some consistency checks. Registers are maintained internally as structure following the C bit notation, where properties are defined for bits. For example, for port 2, we have:

```
struct __p2 {
    unsigned int STATUS : 2;
    unsigned int ERR : 6;
};
```

9 Related work

Although the testing of device drivers is an important concern in most Operating Systems, generic tools and methodologies to do this are surprisingly few. Microsoft provides one such set of tools [2] to perform some performance and compatibility tests on device drivers for Windows.

Devil [6, 7] is a language for generating device driver stubs in a safe and efficient way. Blue Water Systems' WinDK [5] and Compuware's NuMega [1] are other tools to generate device driver stubs. Saint on the other hand, generates stubs for soft devices, which encapsulate the semantics of the device in a limited manner.

10 Conclusion and Future work

We have implemented a limited framework for soft devices. We intend to build on this framework, partly by extending the Saint language to be able to define more complex interfaces, such as PCI, ISA, USB etc.. In the context of Saint, we would also like

to be able to add more functionality so as to be able to describe entire soft devices instead of only interfaces to generate stubs. We would also like to implement some real soft devices to better understand some of the topics we have broached, such as profiling hardware devices. Finally, we would like to explore possibilities of a systematic tool set to be able to evaluate the efficiency of device drivers, and compare the performance of different device drivers using soft devices.

11 Acknowledgments

We would like to thank Jeff Dike, the creator of UML, for helping us resolve some UML related issues during the course of this project and Brian Code and Hedi Hamdi for their useful comments.

References

- [1] Compuware Corporation. Driverwork's user guide. www.numega.com
- [2] Microsoft Corporation. Windows Device Driver Development Kits. www.microsoft.com/whdc/ddk/winddk.mspix
- [3] Jeff Dike. User-mode Linux. In Proceedings of the Ottawa Linux Symposium, 2001
- [4] Jeff Dike. Making Linux safe for virtual machines. In Proceedings of the Ottawa Linux Symposium, 2002
- [5] Blue Water Systems Inc. Windk users manual. www.bluewatersystems.com
- [6] F. Mérillon, L. Réveillère, C. Consel, R. Marlet and G. Muller. Devil: An IDL for Hardware programming. In Proceedings of the fourth symposium on Operating Systems Design and Implementation, pages 17-30, San Diego, California, October 2000
- [7] L. Réveillère and G. Muller. Improving Driver robustness: an evaluation of the devil approach. In The International Conference on Dependable Systems and Networks, pages 131-140, Göteborg, Sweden, July 2001, IEEE Computer Society. ■

Referat fra DKUUG generalforsamling 16/11 2004 i Symbion



Sådan ser bestyrelsen ud | Foto:Flemming Jacobsen

1) Valg af dirigent

Erik Gravgaard blev foreslået af bestyrelsen og blev valgt som dirigent

2) Valg af referent

Henrik Lund Kramshøj blev valgt som referent

3) Beslutning om mødet er lovligt indkaldt

Det blev konstateret at generalforsamlingen er lovligt indkaldt alle formalia er overholdt

4) Behandling af bestyrelsens beretning

Der er ingen spørgsmål til denne. Ulf takker af og fik i den forbindelse gaver.

Udvalg - i alfabetisk rækkefølge

ADM – Sidsel fortalte om ADM

BLD – Sidsel fortalte om dette udvalg og problemerne med portostøtten blandt andet at der vil komme elektroniske udgaver i fremtiden og at de gamle blade vil blive lagt ind i et elektronisk system.

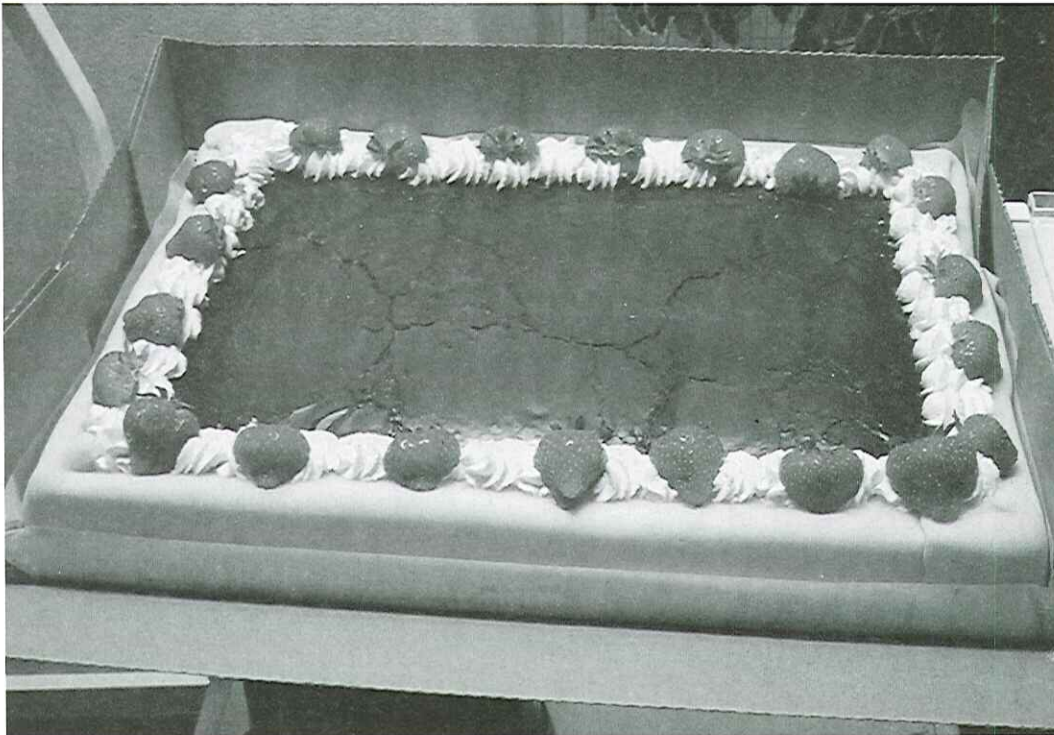
KLB – udvalget Sidsel fortalte om ambitionen med at få gang i klubaftenerne. Der har været en del møder :-)

MMU Sidsel fortsatte med beskrivelsen af dette

udvalg og ideerne med BOF dagen, linuxforum 2004 som var den 7. konference! I år 2005 flytter linuxforum væk fra symbion primært af pladsehensyn. Svært at få økonomien til at hænge sammen Walter Belgers seminaret havde 4 deltagere!

Ext Kristen Nielsen overtog podiet og fortalte om samarbejde med andre foreninger – primært i udlandet. Det nordiske samarbejde og det europæiske samarbejde, hollandske, engelske, tyske og ungarske UNIX brugergrupper. Møderne er afholdt i forbindelse med konferencerne som er afholdt i løbet af året. Her vil DKUUG også gerne digitalisere gamle arkiver af materiale. Planen for 2005 er at EUUG/Europen.org webportal åbnes.

Kristen fortsatte med beskrivelsen af aktiviteterne i NET udvalget. Drift og vedligehold af services i DKUUG. I 2004 er der blandt andet kommet væsentlig mere diskplads på FTP siden og en masse nye mirrors. Der er kommet WLAN på DKUUG's kontor – hvor man skal kende koden. Betalingssystemet er migreret til webserver. Planen er at implementere IPv6 til web og ftp,



Ingen generalforsamling uden kage | Foto: Flemming Jacobsen

indsat ny mailserv og DNS server i driften, ny server på kontoret, etableret en admin og build-server til driften, server til medlems e-services.

NordU2004 konferencen gennemgik Kristen også.
Erik Gravgaard fortalte om marketing udvalget.

Peter Toft fortalte om standardiseringsudvalget. Arbejdet har primært drejet sig om definitionen af hvad åbne standarder er. Der er brugt en del krudt på standardmaskinen og FreeBSD eller ej – og det endte med at det blev FreeBSD. Keld trak herefter en del aktiviteter ud fra denne server/DKUUG - en del ISO organisationer.

Donald Axel overtog herefter og talte om åbne standarder. STD har blandt andet søsat <http://www.aaben-standard.dk>.

Sidsel fortalte om bestyrelsensvisionsdag

5) Godkendelse af regnskab

Erik Gravgaard gennemgik regnskabet
Evan Klarholt forklarede at der var en blank påtegning og gav yderligere information om god regnskabsskik

Regnskabet blev godkendt

Kagepause! 19:28 til 19:58

6) Valg af formand

Kristen Nielsen og Sidsel Jensen opstiller til formandsposten Det blev aftalt at der er ca. 5 min til valgtaler, hver især.

På uret på min Mac blev følgende tider registreret

Sidsel: 20:04 - 20:05 - meget kort

Kristen: 20:05 - 20:09 + en slide :-)

– derefter lidt yderligere snak om bestyrelse og www.thecamp.dk

Sedler indsamlet 20:12

Optælling igang ... 20:14 resultat afleveret til Erik Gravgaard

Samtlige 15 mulige stemmer afgivet og med følgende fordeling:

Sidsel: 9 stemmer

Kristen: 6 stemmer

7) Valg af bestyrelse

Kristen genopstiller



Bestyrelsens beretning 2003/2004

Følgende kandidater opstiller – listen som udsendt:

Rasmus Barfoed
Poul Erik Thamdrup
Peter Makholm
Henrik Lund Kramshøj
Donald Axel

Samtlige er valgt :-)

Der skal ved lodtrækning afgøres hvem der kun er valgt for 1 år – som foretages på det konstituerende møde

8) Valg af revisor og revisorsuppleant

Bo er genvalgt med rungende klapsalver

Lars Josephsen blev (ud)valgt som revisorsuppleant.

9) Fastsættelse af årskontingent

Herunder budget

Budget blev diskuteret som sædvanligt og endeligt godkendt med uændret kontingent.

10) Behandling af øvrige sager foreslået

Det kan konstateres at der på generalforsamlingen ikke er fremmødt nok til at det overhovedet KAN vedtages.

Der er ikke 2/3 af samtlige stemmer til stede.

Erik Gravgaard fremlagde forslaget som er udsendt

Sammensætningen af foreningen gør at der er 77 stemmer og der skal derfor være 52 der stemmer for. På generalforsamlingen er der som tidligere nævnt 15 stemmer.

Det af bestyrelsen forelagte forslag er enstemmigt vedtaget med 14 stemmer - enstemmigt idet en stemme har forladt generalforsamlingen.

11) Eventuelt ■

Vi var en nyvalgt bestyrelse med en hel del nye medlemmer, og vi har derfor brugt en masse tid på at finde den korrekte samarbejdsform.

Kort efter nytår måtte foreningens mangeårige formand Ulf Nielsen af familiemæssige årsager trække sig, og overlade styringen af foreningen til næstformand Peter Toft. Kort efter sommerferien måtte konstitueret formand Peter Toft også af familiemæssige årsager trække sig, hvorefter Ulf Nielsen var tilbage på formandsposten, og Kristen Nielsen blev næstformand.

Den 25. september 2004 holdt bestyrelsen en visionsdag, hvor vi diskuterede vores visioner og prøvede at finde frem til en langsigtet strategi og plan for DKUUGs fremtid. De præliminære resultater vil blive præsenteret på generalforsamlingen.

Bestyrelsen har i indeværende bestyrelsesår taget følgende initiativer:

- Månedlige Nyhedsbreve til orientering af foreningens mange aktiviteter
- Øget samarbejdet med andre foreninger indenfor vores virkeområde
- Langt bedre brug af vores webside.
- Faste månedlige klub-møder
- Udarbejdet forslag til vedtægtsændringer, der giver alle medlemmer stemmeret.

I det kommende år er det den siddende bestyrelses hensigt at:

- Integrere de nye bestyrelsesmedlemmer og foreningens medlemmer i det igangværende visionsarbejde
- Styrke kommunikationskanalen til medlemmerne
- Styrke medlemsadministrationen i sekretariatet (ny medlemsdatabase med selvbetjening)
- Være mere lydhøre overfor medlemmernes ønsker.

Bestyrelsen i DKUUG ■

Stemmeret til alle medlemmer

Det er DKUUGs bestyrelse en stor glæde at kunne annoncere, at de på generalforsamlingen foreslåede vedtægtsændringer nu er en realitet!

Efter foreningens urafstemning sluttede d. 23/12 2004, kunne sekretariatet meddele at resultatet blev følgende:

Antal stemmer for	53
Antal stemmer imod	0
Antal der ikke har stemt	23
Antal afsendte stemmesedler	76

Bestyrelsen vil derfor gerne benytte lejligheden til at opfordre de individuelle medlemmer til at benytte denne enestaaende mulighed for at engagere sig yderligere i foreningens daglige virke.

DKUUG består bl.a. af en række udvalg baseret på frivilligt arbejde, som gerne tager imod nye aktive medlemmer. Det drejer sig bl.a. om:

STD – DKUUGs standardiseringsudvalg
(std@dkuug.dk)

NET – DKUUGs netværksudvalg (net@dkuug.dk)

BLD – Udvalget der udarbejder foreningens blad DKUUG-Nyt (bld@dkuug.dk)

KLB – DKUUGs klubudvalg der arrangerer tekniske aftenmøder (klb@dkuug.dk)

Hvis man ønsker medindflydelse på foreningen DKUUG – er udvalgene der hvor arbejdet laves.

Det nye sæt vedtægter vil blive gjort tilgængelige via foreningens hjemmeside i løbet af kort tid.

Mvh. Bestyrelsen ■



DKUUG NYT

er medlemsbladet for DKUUG,
foreningen for Åbne Systemer
og Internet

UDGIVER

DKUUG
Fruebergvej 3
2100 København Ø
Tlf.: 39 17 99 44
Fax: 39 20 89 48
e-mail
dkuugnyt@dkuug.dk
Sekretariatet er åbent:
Mandag - fredag
kl. 10.00-15.00

REDAKTION

Hanne Vilmann
(ansvarshavende)
Keld Simonsen
Henrik L. Kramshøj
Sidsel Jensen

LAYOUT

Dorte P. grafisk design

TRYK

Palino Print

Annoncer
Kontakt DKUUGs
sekretariat

OPLAG

1300 stk.

Artikler m.v. i DKUUG-Nyt er
ikke nødvendigvis i overens-
stemmelse med redaktionens
eller DKUUGs bestyrelses
synspunkter.
Eftertryk i uddrag med kilde-
angivelse er tilladt.

DEADLINE

Deadline for næste nummer
nr. 151 er den 20. maj 2005

Medlem af Dansk Fagpresse

DKUUG NYT

ISSN 1395-1440



Formandens side

Så er vi kommet godt ind i år 2005 og du sidder med årets første DKUUG-Nyt i hænderne. Som du måske allerede har bemærket er bladet blev shinet grafisk op med et lidt nyere design. Vi håber du kan lide det.

Det er en del af den nye linie, som DKUUG prøver at lægge. Sammen med bladet finder du også en CD med kontorpakken OpenOffice i en lækker indpakning. Det er en gave fra DKUUG til medlemmerne. Du kan vælge at beholde den selv, eller give den til en du tror har brug for CD'en.

Yderligere har bestyrelsen været afsted på visionsdag i februar og skal afsted igen i maj måned. Herefter vil resultaterne af arbejdet med en definition af et klart mål, en vision for fremtiden og bestyrelsens handlingsplan blive præsenteret for medlemmerne.

Som du kan læse andetsteds i bladet blev vores urafstemning gennemført med et positivt resultat. Det er dermed lykkedes at ændre foreningens vedtægter for første gang siden de blev revideret i 1995. Ændringen har resulteret i stemmeret til alle medlemmer i foreningen. Det er bestyrelsens håb at ændringen vil føre til mere aktive medlemmer i foreningen.

2005 tegner sig indtil nu for et spændende år for DKUUG.

Sidsel Jensen
Formand for DKUUG ■

SKRIV TIL

**sek@dkuug.dk,
hvis i mener at i skal
være den næste
»Profil af et medlem«**

The Celebrity Deathmatch – series

AF SIDSEL JENSEN

En celebrity deathmatch? Hvad pokker er det og hvad har det med UNIX at gøre, tænker du sikkert. Ideen udsprang af alle de religionskrige der ud-kæmpes om forskellige programmer, nogle gange med skyttegravslignende forhold. Det må kunne gøres anderledes og mere sobert! Så var det jeg kom til at slå over på MTV og der har de nogle små indslag med nogle fuldstændig fjollede karikaturer af kendisser i modellervoks, som afholder »Celebrity Deathmatches« i en boksering ...

Første arrangement blev afholdt i sommeren/efteråret 2003 (over to gange) med udgangspunkt i de forskellige mailserver implementationer: Qmail, Postfix, Sendmail og Exim. Arrangementet var en kæmpe success med godt 100 fremmødte hver gang. I løbet af dagene efter blev det derfor aftalt, at der skulle laves en version 2.



Dommerpanelet | Foto: Kristian Vilman



Hanne Pathos Vilman, Anders Aramis Søndergaard og Peter Athos | Foto: Peter Toft

I april i år var det tid til Celebrity Deathmatch igen. Denne gang var emnet de forskellige kommando-fortolkere: sh, bash, ksh, zsh, Plan 9s rc og cmd.exe. Lokalet M2 på Symbion var proppet til randen med tilhørere da arrangementet blev fløjet i gang i bedste foldboldstil af de tre overdommere, der repræsenterede de tre foreninger DKUUG, SSLUG og BSD-DK, der i fællesskab havde arrangeret aftenens »battle«. Det positive rygte fra version 1, var løbet i forvejen.

Kendetegnet ved Celebrity Deathmatch'ene er at emnerne diskuteres sobert og professionelt, men stadig med et glimt i øjet hvor der både er plads til at chikanere modkompetanterne og lefle for overdommerne (der i bedste stil er totalt forudindtaget). Fordelen for tilhørerne er et underholdende arrangement, der giver et godt og bredt overblik over centrale UNIX relevante emner.

Vinderen af Shell Celebrity Deatchmatchen blev Jesper Louis Andersen fra BSD-DK med ksh. Jesper fik en flaske champagne og en OpenBSD bog som præmie, for sit foredrag. Jesper havde bl.a. lagt små subliminale beskeder ind i sit foredrag med hints som: »David Korn er gud«, »INGEN BLOAT« og »I bliver alle hjernevaskede«.



Mon de finder en vinder? | Foto: Peter Toft

Han var ellers i skarp konkurrence med både Peter Makholms bash fremlæggelse, der inkluderede Belgisk øl til dommerpanelet og Thomas Ammitz-bøll-Bachs 10 bud om bourneshellen:



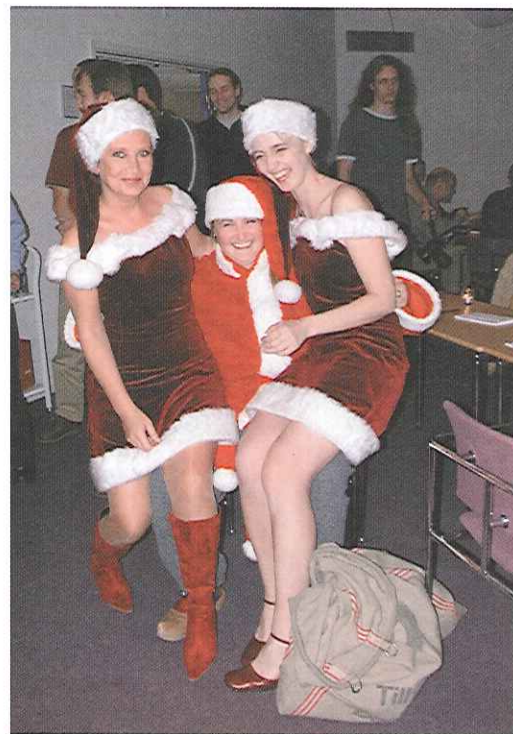
Bourne-Shellen er den oprindelige shell ...

1. root må ikke have andre shells end den
2. Du må ikke symlinke eller hardlinke anden shell end sh fra /bin/sh
3. Seks dage om ugen må du bruge hvilken som helst shell du vil til dit almindelige arbejde, men når du på den syvende dag skal lave systemarbejde i single-user mode, skal din shell være sh
4. Du skal lade dine afledte shell-kloner respektere den generelle syntaks i sh, og ikke som andre opfinde nye syntakser
5. Du må ikke fjerne sh fra filtræet
6. Du må ikke compile sh med dynamisk linking
7. Du må ikke flytte sh til et andet filsystem end rod-filsystemet
8. Du må ikke omdøbe en anden shell til /bin/sh
9. Du må ikke begære features i sh fra andre shells
10. Du må ikke begære features i sh fra andre programmeringssprog, libraries eller moduler.

Torsdag d. 16 december 2004, blev version 3 afholdt. Denne gang var emnet de forskellige UNIX editorer: ed, vim, vigor, emacs og Plan 9s acme. Poul-Henning Kamp lagde for med et foredrag præsenteret i ed, scrollende i 300 baud hastighed – stilen var dermed lagt for aftenen! Dommerne var

som sædvanligt i højt humør – julenisse-humør – og lokalet M1 var julepyntet efter alle kunstens regler. Som ved de to øvrige deathmatcher var lokalet fyldt med interesserede lyttere, der alle fik sig en munter aften i hyggen og teknikkens navn. Aftens vinder blev ikke mindre end Anders Søndergaard med editoren ACME fra Bell Labs Plan 9 styresystem.

Editoren høstede bl.a. tilråb fra publikum, som »Det er jo altsammen meget godt, men kan man også skrive tekst i den?« - idet Anders i sin iver for at vise forskellige features i sin demo fuldstændig havde glemt at vise noget så banalt som indsætning af tekst.



Mon de får hvad de ønsker? | Foto: Thomas A. Frederiksen

Jeg kan på det kraftigste opfordre til, at man dukker op til næste deathmatch, for nu skriver vi 2005, så mon ikke der bliver en deathmatch eller 2 i år? Der er jo stadig mange spændende emner at tage fat på. Hold øje med www.dkuug.dk for annoncering eller meld dig på DKUUGs annoncerings-liste <seminar@dkuug.dk>. ■

GEAR UP!

DANMARKS SEJESTE GADGETMAGASIN

GEAR



I GEAR finder du alt om det nye, uimodståelige digitale legetøj:

Håndholdte computere med indbygget tv, armbåndsure med kamera og email-boks, verdens lækreste hovedtelefoner, dvd-afspillere, der husker dine ynglingsfilm, notebooks i pristest, farvetv, der brygger kaffe – og meget andet til legedrenge i alle aldre

GEAR

Bestil abonnement på www.ncom.dk/cam eller tlf. 7023 3500

INFO

Udkommer: 12 gange pr. år

Løssalgspris: 39,95 kr.

Abonnement: 399,- kr. (12 mdr.)

Sidetæl: 84-100 sider

Indhold: I GEAR finder du alt om det nye, uimodståelige digitale legetøj: Håndholdte computere, mobiltelefoner, DVD-afspillere, hi-fi, digitale kameraer, notebooks og meget andet til legedrenge i alle aldre.



Normalpris 778,-
Spar 379,-
Din pris 399,-*



399,-*

* der pålægges 25 kr. porto – så længe lager haves.

12x

GEAR

BESTIL NYT 376-SIDERS 2005 GRATIS KURSUSKATALOG PÅ
KATALOG@SUPERUSERS.DK ELLER TLF. 48 28 07 06

SUPERUSERS

2005

Danmarks største
it-kursuskatalog



Kurser · Konsulenttydelser · Certificering
Linux/UNIX, Windows, NetWare
Netværk, TCP/IP, VoIP, VPN, XML
C/C++, C#, Java, Perl, VB.NET, SQL