

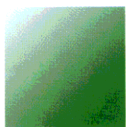
AVANCERET SOFTWARE RESCUE, SIDE 4

OPEN SOURCE DAYS - TALER, SIDE 14

OPEN SOURCE DAYS - KURSER, SIDE 18



Open Source Days 2012



BREV FRA REDAKTØREN

Open Source programmører har fået erfaringer gennem mere end 15 år, traditionen går flere årtier tilbage. Allerede for 31 år siden konstaterede Time Magazine at computer-industrien i modsætning til andre industrier var villig til at dele opfindelser, som måske var patenterbare. Time Magazine gjorde IBM-PC'en til "årets mand" og havde en slags statue på forsiden. Inde i bladet var der artikler om hvordan en hard-disk på 10 MB kunne hjælpe en mindre produktions-virksomhed til bedre planlægning.

I samme periode beskrev Time, hvordan produktion af passagerfly blev økonomisk fordi man de tusindvis af hydrauliske rør, som indgår i styringen af et fly, kunne fremstilles computerstyret, når der var brug for dem. Reserveredslageret kunne reduceres. Det var succes-historierne, men der var også alvorlige svipsere.

Måske nogle af læserne er af min årgang og husker, hvordan ledelsen i DR og Christian Roving umodent tog fat på en slags computerstyret video-redigering, DORA, på et tidspunkt hvor hverken grafiske enheder eller CPU'er var hurtige nok, hvordan den ene skandale efter den anden udspillede sig - og stadig udspilles - fordi ledere og planlæggere ikke har hands-on fornemmelse for hvad teknikken egentlig kan.

Det er meget fascinerende at se, hvordan Open Source guruer har grebet problemerne an i projekt management.

Open Source har oplevet de samme op- og nedture, og graden af kompleksitet kan ikke øges yderligere uden bedre projektstyring, herunder code-management. Linus Torvald har skabt program-suiten Git - et skalerbart, distribueret

revisions-kontrol system. Det er blevet brugt på mange forskellige måder, og det kan man høre om på Open Source Days (tidligere LinuxForum) konferencen 10-11.marts.

Kurser afholdes i de to dage før selve konferencen, bl.a. i Application Lifecycle Management with Git, Gerrit & Jenkins.

Foredragene spænder vidt:

- Webkit, Why it Matters
- Modern Web Programming in Perl
- Mass Deployment of Unix Machines
- Occupy Wall Street's Information Tech: Facebook For The 99%,
- An Update on Open Source Business Models and Issues,
- QT project and QT5,
- Interesting QT5 Addons dn QT5 based solutions
- Why Joomla?
- NoSQL - Databases without SQL,
- Arduino Prototyping Platform, Embedded Electronics (and more).

Men ved redaktionens afslutning kom der stadig flere emner på programmet til OSD, så læserne bedes besøge www.opensourcedays.org (eller linuxforum.dk) og se, hvordan programmet bliver.



Det kan være hårdt med to dages Open Source seminar. Så er det rart at slappe af og indtage lidt godt til maven i Symbions hyggelige kantine.

Indhold i dette nummer

I bladet her har vi denne gang artikler om hardware, systemadministration, programmeringssprog. Foruden om-tale af OSD bringer vi også referat fra generalforsamlingen i november. Jeg oplevede det som den første fredelige generalforsamling i de 8 år jeg har deltaget, men da jeg talte med kollegerne i bestyrelsen blev jeg gjort opmærksom på at der faktisk var protester under selve generalforsamlingen. De mere alvorlige ting tidligere må have hærdet mine ører. Eller også er jeg ved at blive døv.

Referatet findes bagest i bladet.

Desuden har økonomien bedret sig lidt, så vi kan trykke net-bladet fra November 2011. Heri findes en artikel om IPv6 tunneling på en nem måde.

Nye forslag

DKUUG management ændrer sig, og brandet ændrer sig. Her på redaktionen mener vi at DKUUG står for Danish Kingdom of United Unix Gurus - eller er der noget vi har misforstået? Der er også forslag om at DKUUG skal være Dansk Open Source/Standards Users - DOSSU. Det er vist et lidt mere seriøst bud. Har du nogen forslag?

Deadline for næste nummer er 15.maj 2012, og vi håber på inspirerende input.

Redaktionen
Donald Axel

INDHOLD:



DKUUG-Nyt er medlemsbladet for DKUUG, foreningen for Åbne Systemer og Internet

Udgiver:

DKUUG
Fruebjergvej 3,
2100 København Ø
Tlf: 39 17 99 44
Fax: 39 20 89 48
email: dkuugnyt@dkuug.dk

Redaktion:

Donald Axel (ansvarshavende)
Jon Bendtsen

Forsidefoto:

Donald Axel
OSD på ITU marts 2010

Tryk:

Glumsø Bogtrykkeri A/S

Design og Layout:

Seifert Design & Grafik

Annoncer:

Kontakt DKUUGs sekretariat
sek@dkuug.dk

Oplag:

600 eksemplarer

Artikler m.v. i DKUUG-Nyt er ikke nødvendigvis i overensstemmelse med redaktionens eller DKUUGs bestyrelses synspunkter.

Eftertryk i uddrag med kildeangivelse er tilladt.

Medlem af Dansk Fagpresse

DKUUG-Nyt
ISSN 1395-1440



BREV FRA REDAKTØREN	2
AVANCERET SOFTWARE RAID RESCUE	4
INTRODUKTION TIL LISP.....	8
4 NOTES IN 2U	12
OPEN SOURCE DAYS – TALER V. REDAKTIONENS SLUTNING . . .	14
OPEN SOURCE DAYS – KURSER TORS 8. – FRE 9. MARTS	18
LISP – PROGRAMMERINGSSPROG	22
REFERAT AF DKUUG GENERALFORSAMLING NOV. 2011	23

KALENDER:

Open Source Days, lø-sø 10.-11. marts 2012. Start lø kl.9:00, sø kl.9:00. Slut ca. kl. 17:00

Adresse: Copenhagen Business School (CBS), Solbjerg Plads 3, 2000 Frederiksberg.

Bus linier: 18 ad Falkoner Alle til Frederiksberg Centeret og CBS, 4a til Fasanvej Station, 14 fra Vesterport Station, 15 fra Hovedbanegården. Køreplaner kan downloades fra Movia.dk

Metro: M1 og M2 til Frederiksberg Station. Metro-hjemmeside: m.dk

Offentlig transport i København er rimelig god. Transport i bil kan også være effektiv især udenfor myldretiderne. Taxi er dyrt. Generelt anbefaler vi offentlig transport (Metro).

Pris- og billetsystem for Offentlig transport: København er delt i zoner, CBS er i zone-1.

Billetter til 2-3-4 zoner m.v. gælder både bus, S-tog og metro m.v. Billet kan købes med smartphone.

Parkering kan være vanskelig at finde, regn med at bruge et kvarter eller mere på parkering.

Bedst via Nylandsvej, Solbjerg Plads.

ØVRIGE ARRANGEMENTER:

DKUUG foredrag og hyggeaftener i DKUUG's lokaler, Symbion, Fruebjergvej 3, 2100 København:

Torsdag 10. maj kl. 18: IPv6 routing problemer og politik - ikke et kursus i HW for routing, men et IT-politisk overblik. Donald Axel og Jon Bendtsen. Tilmelding nødvendig.

Torsdag 24. maj kl. 18: DKUUG bladredaktion workshop, hvordan man skriver teknisk og sagligt, med et sideblik til stil og ordvalg i klassisk litteratur for programmører samt et blik på Git som filesharing platform. Donald Axel

Torsdag 7. juni kl. 18: DKUUG hyggeaften, Gustav & andre lægger ud med en sang.

THE CAMP:

I 2012 præsenterer vi for 11. gang en do-IT-yourself-sommerlejr for computernørder.

Vi har rammerne - landlig idyl med internetforbindelse, strøm nok og en hel uge til at lave præcis det vi har lyst til. Lejren foregår i uge 30, og varer fra lørdag den 21. juli til lørdag den 28. juli 2012. Billetter sælges via billetnet og vi forventer at salget starter i starten af marts 2012.

AVANCERET SOFTWARE RAID RESCUE



Jon Bendtsen er indehaver af Jonix.dk hvorfra han leverer IT konsulent-ydelser. Jon Bendtsen har beskæftiget sig med IT siden midten af 90'erne og er uddannet datalog fra DIKU, Datalogisk Institut, Københavns Universitet.

AF JON BENDTSEN.

En ny kunde kontaktede mig efter at deres server var gået ned og ikke kom pænt op igen. Det viste sig at strømmen var gået to gange – og anden gang var den gået midt i et software raid resync.

De havde et større setup med mange video data som ville tage flere dage at restore, hvor imens de ikke ville kunne arbejde, så jeg måtte jo se hvad jeg kunne gøre for at fixe deres server.

Jeg loggede ind som root og begyndte at undersøge problemet. Kommandoerne, mount, pvscan, vgdisplay, lvdisplay, cat /proc/mdstat samt cat /etc/fstab plejer at afsløre hvor de fleste data ligger.

Kommandoen pvscan i **data box 1** afslørede da også at der var en volumen group, men at den manglede et device, som sikkert ville være et software raid, for det var de 2 andre devices.

```
video:/root# pvscan
Couldn't find device with uuid Zzd7hA-60yf-Icay-udC9-niLY-7VSI-yCM2ih.
PV /dev/md0      VG VG    lvm2 [5.46 TiB / 0 free]
PV /dev/md1      VG VG    lvm2 [5.46 TiB / 0 free]
PV unknown device VG VG    lvm2 [7.28 TiB / 0 free]
Total: 3 [18.19 TiB] / in use: 3 [18.19 TiB] / in no VG: 0 [0 ]
```

DATA BOX 1

```
Personalities : [raid1] [raid6] [raid5] [raid4]
md2 : inactive sdd1[0](S) sdu1[6](S) sdw1[4](S) sdm1[2](S) sdc1[1](S)
      9767554916 blocks super 1.2

md1 : active (auto-read-only) raid6 sde1[0] sdk1[7] sdl1[6] sdq1[5]
      sdpl[4] sdjl[3] sdgl[2] sdvl[1]
      5859306240 blocks level 6, 64k chunk, algorithm 2 [8/8] [UUUUUUUU]

md5 : active raid1 sda2[0] sdb2[1]
      4008128 blocks [2/2] [UU]

md4 : active raid1 sda1[0] sdb1[1]
      2008000 blocks [2/2] [UU]

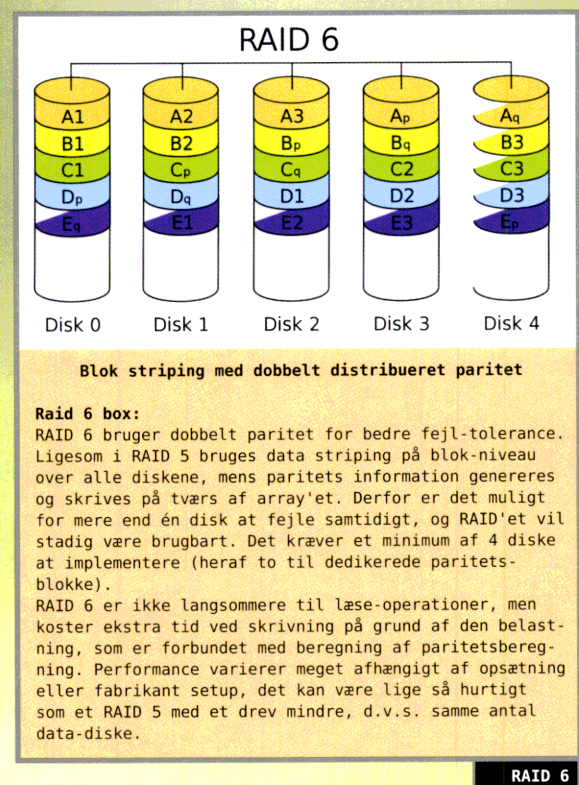
md0 : active (auto-read-only) raid6 sdf1[15] sdt1[9] sdol[10] sdil[8]
      sdr1[11] sdh1[12] sds1[13] sdn1[14]
      5860558848 blocks super 1.2 level 6, 64k chunk, algorithm 2 [8/8]
      [UUUUUUUUU]

unused devices: <none>
```

DATA BOX 2

Kommandoen **cat /proc/mdstat** afslører i **data box 2** at det ene raid var inaktivt, og kikker man nærmere på det i **data box 3** med kommandoen **mdadm --detail /dev/md2** så ser man, at det inaktive software raid består af 6 diske som kører *raid level 6* (se beskrivelse og illustration nedenfor).

Raid level 6 skal kunne klare at miste 2 vilkårlige diske, og dette raid havde jo netop kun mistet to diske. Hvorfor virkede det ikke? Det skulle det jo, ifølge teorien.



RAID 6

Jeg havde på fornemmelsen at de enkelte diske måtte være ude af sync, men håbede at **--run og --force** kommandoerne fra **cmd box 1** ville tvinge raidet til at starte alligevel, således at jeg i det mindste


```
video:/root# mdadm --detail /dev/md2
/dev/md2:
  Version : 1.2
  Creation Time : Fri Apr 1 16:06:03 2011
  Raid Level : raid6
  Used Dev Size : 1953510400 (1863.01 GiB 2000.39 GB)
  Raid Devices : 6
  Total Devices : 5
  Persistence : Superblock is persistent

  Update Time : Thu Apr 14 04:56:57 2011
  State : active, degraded, Not Started
  Active Devices : 4
  Working Devices : 5
  Failed Devices : 0
  Spare Devices : 1

  Layout : left-symmetric
  Chunk Size : 512K

  Name : video:2 (local to host video)
  UUID : 75bd628c:49f5e61c:24f82a40:69ce5bff
  Events : 1968

Number Major Minor RaidDevice State
  0       8       49        0   active sync  /dev/sdd1
  1       8       33        1   active sync  /dev/sdc1
  2       8      193        2   active sync  /dev/sdm1
  3       0        0        3   removed
  4      65       97        4   active sync  /dev/sdw1
  5       0        0        5   removed
```

DATA BOX 3

```
video:/root# mdadm --examine /dev/sdc1
/dev/sdc1:
  Magic : a92b4efc
  Version : 1.2
  Feature Map : 0x0
  Array UUID : 75bd628c:49f5e61c:24f82a40:69ce5bff
  Name : video:2 (local to host video)
  Creation Time : Fri Apr 1 16:06:03 2011
  Raid Level : raid6
  Raid Devices : 6

  Avail Dev Size : 3907021954 (1863.01 GiB 2000.40 GB)
  Array Size : 15628083200 (7452.05 GiB 8001.58 GB)
  Used Dev Size : 3907020800 (1863.01 GiB 2000.39 GB)
  Data Offset : 2048 sectors
  Super Offset : 8 sectors
  State : clean
  Device UUID : 9f3a810d:14e04a6d:55f4c224:3c1bcbb4

  Update Time : Thu Apr 14 04:56:57 2011
  Checksum : 568d5c29 - correct
  Events : 1968

  Layout : left-symmetric
  Chunk Size : 512K
```

DATA BOX 4

kunne forsøge at læse data read-only, så kunne jeg fx måske have set om filesystemet også var smadret, eller kun det ene software raid.

Tvang virkede ikke, så jeg måtte lave en **mdadm --examine** på hver disk i det fejlede raid, men jeg viser kun et eksempel for en enkelt disk (se **data box 4**).

Med så store mængder output er det ikke nemt at se forskelle med det blotte øje, men heldigvis kan man koncentrere sig om Uptime Time, Checksum og Events. Så dem greppede jeg efter.

Her viste det sig at Update Time var forskellig for den ene disk, men den disk var den 6. disk som var helt smidt ud af

raidet (data box 5). Alle Checksummer var korrekte, men forskellige (data box 6). Events feltet i data box 7 afslørede dog tydeligt hvorfor software raidet ikke ville starte – der var 3 forskellige værdier fordelt på de 6 diske i klumper af 3, 2 og 1 disk.

Heldigvis er manualen aldrig langt væk i *nix systemer, så selvom at jeg har læst man-siden for mdadm flere gange og bruger mdadm regelmæssigt for at oprette, udvide eller reparere raids, så kan jeg den ikke udenad. Men jeg kunne dog huske at der eksisterede både en **--assume-clean** og en "missing" option, så et hurtigt kik på man-siden afslørede

hvordan. Det afslørede også, at der er 2 metoder at lave et software raid: *build* og *create* (se **man box 1**). Specielt *build* lød meget lovende, siden den ikke ville ændre på selve diskene, hvilket passede mig fint, for så kunne jeg forsætte arbejdet med at redde data, hvis *build* ikke virkede.

Men inden jeg kunne begynde med *build* måtte jeg lige vide i hvilken rækkefølge diskene indgik i raidet, for ellers så ville de enkelte chunks på 512K komme i forkert rækkefølge, idet *software raid level 6* som default er opbygget som en stor disk med layout

```
video:/root# mdadm --stop /dev/md2
mdadm: stopped /dev/md2
video:/root# mdadm --assemble --uuid=75bd628c:49f5e61c:24f82a40:69ce5bff --run /dev/md2
mdadm: failed to RUN_ARRAY /dev/md2: Input/output error
mdadm: Not enough devices to start the array while not clean - consider --force.
video:/root# mdadm --stop /dev/md2
mdadm: stopped /dev/md2
video:/root# mdadm --assemble /dev/md2 /dev/sdc1 /dev/sdd1 /dev/sdu1 /dev/sdm1 /dev/sdw1
mdadm: /dev/md2 assembled from 4 drives and 1 spare - not enough to start the array while not clean - consider --force.
video:/root# mdadm --run /dev/md2
mdadm: failed to run array /dev/md2: Input/output error
video:/root# mdadm --run --force /dev/md2
mdadm: failed to run array /dev/md2: Input/output error
video:/root# mdadm --stop /dev/md2
mdadm: stopped /dev/md2
video:/root# mdadm --assemble --uuid=75bd628c:49f5e61c:24f82a40:69ce5bff --run --force /dev/md2
mdadm: failed to RUN_ARRAY /dev/md2: Input/output error
```

CMD BOX 1


```
video:/root# for disk in $(fdisk -l | grep 1953510400 | awk '{print $1}'); do echo -n $disk ; mdadm --examine $disk | grep "Update Time"; done
/dev/sdc1    Update Time : Thu Apr 14 04:56:57 2011
/dev/sdd1    Update Time : Thu Apr 14 04:56:57 2011
/dev/sdm1    Update Time : Thu Apr 14 04:56:57 2011
/dev/sdu1    Update Time : Thu Apr 14 04:56:57 2011
/dev/sdw1    Update Time : Thu Apr 14 04:56:57 2011
/dev/sdx1    Update Time : Thu Apr 14 04:55:36 2011
```

DATA BOX 5

```
video:/root# for disk in $(fdisk -l | grep 1953510400 | awk '{print $1}'); do echo -n $disk ; mdadm --examine $disk | grep Checksum; done
/dev/sdc1    Checksum : 568d5c29 - correct
/dev/sdd1    Checksum : d1d4d1cd - correct
/dev/sdm1    Checksum : 219d83b5 - correct
/dev/sdu1    Checksum : 8745e91 - correct
/dev/sdw1    Checksum : f7a8e066 - correct
/dev/sdx1    Checksum : 8aaead0 - correct
```

DATA BOX 6

```
video:/root# for disk in $(fdisk -l | grep 1953510400 | awk '{print $1}'); do
> echo -n $disk
> mdadm --examine $disk | grep Events
> done
/dev/sdc1    Events : 1968
/dev/sdd1    Events : 1968
/dev/sdm1    Events : 1969
/dev/sdu1    Events : 1968
/dev/sdw1    Events : 1969
/dev/sdx1    Events : 1967
```

DATA BOX 7

som i figuren fra wikipedia (Creative Commons copyright Colin M.L. Burnett.) Det gør ingen forskel at jeg har 6 diske og Colins eksempel kun er på 5 diske.

Heldigvis så afslørede **grep** hurtigt i **data box 8** både i hvilken rækkefølge jeg skulle bygge et nyt raid, og at jeg kun havde én måde at gøre det på, for man kan ikke bygge et raid på spare diske.

Rækkefølgen styres simpelthen ved at skrive diskens navn på dens plads, og så ellers skrive kode-ordet "missing" der hvor der ikke skal være nogen disk.

```
Data box 8.
video:/root# for disk in $(fdisk -l | grep 1953510400 | awk '{print $1}'); do
> echo -n $disk
> mdadm --examine $disk | grep Events
> done
/dev/sdc1    Device Role : Active device 1
/dev/sdd1    Device Role : Active device 0
/dev/sdm1    Device Role : Active device 2
/dev/sdu1    Device Role : spare
/dev/sdw1    Device Role : Active device 4
/dev/sdx1    Device Role : spare
```

DATA BOX 8

Jeg prøvede `mdadm -build`, se **cmd build box 1**, men det ikke virker for raid level 6.

Den eneste tilbageværende mulighed

Build Build an array that doesn't have per-device metadata (superblocks). For these sorts of arrays, mdadm cannot differentiate between initial creation and subsequent assembly of an array. It also cannot perform any checks that appropriate components have been requested. Because of this, the Build mode should only be used together with a complete understanding of what you are doing.

Create Create a new array with per-device metadata (superblocks). Appropriate metadata is written to each device, and then the array comprising those devices is activated. A 'resync' process is started to make sure that the array is consistent (e.g. both sides of a mirror contain the same data) but the content of the device is left otherwise untouched. The array can be used as soon as it has been created. There is no need to wait for the initial resync to finish.

Man box 1 - uddrag af manual page for mdadm kommandoen


```
video:~# mdadm --stop /dev/md2
mdadm: stopped /dev/md2
video:~# mdadm --build -n 6 -l 6 --assume-clean /dev/md2 /dev/sdd1 /dev/sdc1 /dev/sdm1 missing /dev/sdw1 missing
mdadm: Raid level 6 not permitted with --build.
```

CMD BUILD BOX - --BUILD KOMMANDO DUER IKKE TIL RAID 6

var **mdadm --create** som ifølge man-siden vil skrive en ny superblok på hver disk, og min nye superblok skulle ligge på præcis samme adresse som den gamle. Den oplysning står i feltet *Version*, og der stod heldigvis den nyeste, version 1.2

Jeg drøftede med kunden om jeg skulle tage en backup af diskene og/eller den gamle raids superblok, før jeg ændrede i dem. Kunden mente at en fuld backup ville tage for lang tid, og de havde jo en rigtig backup de kunne restore fra.

Jeg mente ikke der ville være no-

gen grund til at tage en backup af den gamle superblok, for det var jo tydeligt at den ikke virkede, og jeg kunne ikke se flere muligheder med at få **mdadm** til at starte software raidet.

Nu kom det afgørende øjeblik.

Jeg kørte nu **--create** kommandoen, som den ses i **cmd create box 1**, og fik startet det raid som kan ses i **data box 9, 10 og 11**. Det så godt ud, LVM genkendte raidet, så jeg kunne lige lave en **fsck**, som helt utroligt ikke fandt nogen fejl, og derefter var det jo bare at

mounte raidet. Alle data var reddet takket være **--assume-clean** og **missing**.

- HA! prøv bare at få dit hardware raid til at gøre det samme og være lige så fleksibel -

Herefter var det helt normal procedure at putte de 2 andre diske ind i raidet og lade det resynce som normalt imens kundens ansatte kunne arbejde videre.

```
video:~# mdadm --stop /dev/md2
video:~# mdadm --create -n 6 -l 6 --assume-clean /dev/md2 /dev/sdd1 /dev/sdc1 /dev/sdm1 missing /dev/sdw1 missing
mdadm: /dev/sdd1 appears to be part of a raid array:
  level=raid6 devices=6 ctime=Fri Apr 1 16:06:03 2011
mdadm: /dev/sdc1 appears to be part of a raid array:
  level=raid6 devices=6 ctime=Fri Apr 1 16:06:03 2011
mdadm: /dev/sdm1 appears to be part of a raid array:
  level=raid6 devices=6 ctime=Fri Apr 1 16:06:03 2011
mdadm: /dev/sdw1 appears to be part of a raid array:
  level=raid6 devices=6 ctime=Fri Apr 1 16:06:03 2011
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md2 started.
```

CMD CREATE BOX 1

```
Data box 9.
video:~# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md2 : active raid6 sdw1[4] sdm1[2] sdc1[1] sdd1[0]
      7814041600 blocks super 1.2 level 6, 512k chunk, algorithm 2 [6/4] [UUU_U_]
md1 : active (auto-read-only) raid6 sde1[0] sdk1[7] sdl1[6] sdq1[5] sdp1[4] sdj1[3] sdg1[2] sdv1[1]
      5859306240 blocks level 6, 64k chunk, algorithm 2 [8/8] [UUUUUUUU]
md5 : active raid1 sda2[0] sdb2[1]
      4008128 blocks [2/2] [UU]
md4 : active raid1 sda1[0] sdb1[1]
      2008000 blocks [2/2] [UU]
md0 : active raid6 sdf1[15] sdt1[9] sdol1[10] sdil1[8] sdr1[11] sdh1[12] sds1[13] sdn1[14]
      5860558848 blocks super 1.2 level 6, 64k chunk, algorithm 2 [8/8] [UUUUUUUU]

unused devices: <none>
```

DATA BOX 9

INTRODUKTION TIL LISP

AF DAVID ASKIRK FOTEL



David Askirk Fotel

Lisp er et sprog, som er klart undervurderet i tiden, men det er dog et sprog, som indeholder elementer, der har gjort det berømt gennem de 54 år der er gået, siden det blev skabt i 1958 som en praktisk matematisk notation af Alonzo Church's Lambda kalkyle.

Det blev hurtigt det foretrukne programmeringssprog for forskning i kunstig intelligens.

Foruden at være et af de første programmeringssprog var Lisp tidligt ude med nye begreber og metoder som fx. organisering af data i træstrukturer, automatisk lager-vedligeholdelse eller garbage collection, dynamisk typesætning af data, og en kompiler, som kan kompilere sig selv.

Meget kort lisp historie (fra <http://da.wikipedia.org/wiki/Lisp>):

"Sproget er baseret på en matematisk teori - Alonzo Church's lambdakalkyle (lambda calculus) fra 1936. Det blev specificeret af John McCarthy i 1958. Beskrivelsen blev publiceret i 1960 og sproget blev for første gang implementeret af [5]Steve Russell i 1961. Dermed er LISP et af de ældste højniveau-programmeringssprog; blandt de nulevende og stadigvæk mest udbredte er det kun Fortran der er ældre."

Jeg vil i denne artikel give en kort introduktion til lisp ved brug af common lisp implementationen kaldet **Steel Bank Common Lisp** (<http://www.sbcl.org/>).

Vi starter med hello world og vil slutte af med et lille spil.

Men som når man lærer andre sprog, må vi hellere begynde med begyndelsen.

```
Kommandolinie start af sbcl
herosan-2:~ davse$ sbcl
This is SBCL 1.0.47, an implementation of ANSI Common Lisp.
More information about SBCL is available at <http://www.sbcl.org/>.
SBCL is free software, provided as is, with absolutely no warranty.
It is mostly in the public domain; some portions are provided under
BSD-style licenses. See the CREDITS and COPYING files in the
distribution for more information.
*
```

BOX 1

Først skal **sbcl** installeres, det gøres nemmest via en pakkemanager eller *mac ports*. (fx. Debian-derivater: **apt get install sbcl** eller på Red Hat derivater: **yum install sbcl** eller **yum install clisp**)

Når sbcl (eller lignende) så er installeret, er det bare at starte det; skriv **sbcl** på kommandolinien, så starter Lisp fortolkeren forhåbentlig (se box 1). Den udprinter en asterisk (en stjerne) (eller i clisp en tæller og en vinkel).

Nu står cursoren i lisps REPL, Read Evaluate Print Loop. Den er klar til at modtage kommandoer, ligesom python er, når man bruger interaktiv python.

Lad os starte med hello world:

```
* (format t "Hello World~%")
Hello World
NIL
*
```

Så skriver den pænt tilbage til os, og melder at REPL'en er klar igen.

Vi har brugt kommandoen (*format*) til at skrive ud på skærmen. Den kan mange af de samme ting, hvis ikke flere, som printf i C. Samtidig kan vi også se hvordan vi kalder funktioner i Lisp.

Alt i Lisp er lister, også funktions kald. En liste i Lisp starter med "(" og slutter med ")".

Så vi beder REPL'en om at først læse listen (*format t "Hello World~%"*), derefter evaluere den, og til sidst udskrive resultatet.

Når den bliver evalueret sker der også det, at den udskriver *Hello World* til os.

Funktionen (*format*) returnerer *nil*, som så er det næste at REPL'en udskriver til brugeren. **Nil** - ingenting. Det er OK for vores formål lige nu, vi har ikke brug for at Lisp returnerer en værdi. Derefter er REPL'en igen klar til brug.

Men lad os kigge lidt mere på hvad vi kan bruge (*format*) til:


```
* (defvar *besked* "Hello World")
*BESKED*
* (format t "~a~%" *besked*)
Hello World
NIL
*
```

BOX 2

Her laver vi et klassisk trick med at gemme beskeden i en streng variabel og derefter udskrive den. ~% svarer til \n og ~a beder bare om at vi sætter indholdet ind som det er, dvs en liste vil blive udskrevet som en liste, og en streng som en streng.

Lad os lave flere variable: Her er *a* en integer, *b* en streng og *c* er en liste. I Lisp er det normalt at globale variable bliver givet på formen *navn*.

```
box3: Flere variable
*(defvar *a* 42)
*A*
*(defvar *b* "Hi World")
*B*
*(defvar *c* '(1 2 3))
*C*
*(format t "~a~%" *a*)
* 42
NIL
*(format t "~a~%" *b*)
* Hi World
NIL
*(format t "~a~%" *c*)
* (1 2 3)
NIL
```

BOX 3

Når vi udskriver det, bliver listen printet som ascii-repræsentation af elementernes værdier, og elementerne kan være tal eller strenge.

Det ser jo ganske fint ud, så lad os hoppe videre til noget mere spændende.

Sand Falsk

I Lisp er t true og nil false. nil er også den tomme liste, så () er det samme som nil, og derved falsk.

Programsource i en fil

Det næste eksempel er lavet i en fil. Man loader en fil ind i REPL'en med en linie, der ser sådan ud: **(load)**. En linie, der begynder med semikolon, er en kommentar:

```
;File: truefalse.lisp
;True false and if
(defvar *trueOrFalse* t)
(format t "~a~%" *trueOrFalse*)
(setq *trueOrFalse* nil)
(format t "~a~%" *trueOrFalse*)(setq *trueOrFalse* t)
(if *trueOrFalse*
  (format t "Jeg er sand~%")
  (format t "Jeg er falsk~%"))
(defvar *d* 2)
(defvar *e* 2)
(if (= *d* *e*)
  (format t "They are the same~%")
  (if (< *d* *e*)
    (format t "d is less than e~%")
    (format t "d is greater than e~%")))
;Or cond:
(cond
  ((= *d* *e*) (format t "They are the same ~%"))
  (< *d* *e*) (format t "d is less than e~%"))
  (t ("d is greater than e~%"))
```

BOX 4

Når vi kører ovenstående får vi flg. skærmoutput:

```
* (load "truefalse.lisp")
T
NIL
Jeg er sand
They are the same
They are the same
T
*
```

Programmets printer (viser/tegner) stjernen, som betyder at den er klar til vores input. Vi skriver **(load "filnavn")** og derved loader vi filen ind. Den bliver kørt med det samme, da kommandoerne står direkte i filen. De to første format, udskriver variabelen. Værdien af variabelen bliver ændret med **(setq)**.

Derefter har vi den første if. Her kan man se at **(=)**, som er en af sammenligningsoperatorerne, står forrest i den liste af elementer som den sammenligner. Man kan nemlig i Lisp lave kald som **(= a b c)** eller flere variable, hvis det ønskes.

if er bygget op så den får to lister. Den ene liste er de kommandoer der bliver kørt hvis udtrykket er sand, den anden hvis det er falsk.

Den næste bid viser nestede if sætninger. Den sidste del af filen viser en **(cond)** som er en funktion der gør det nemmere og pænere at evaluere mange udtryk efter hinanden. Der er altid en default action som bliver udført hvis de andre fejler.

Men Lisp er jo også funktionalt, udover proceduralt og object orienteret. Det er multiparadigme. Så her kommer lige lidt forskellige måder at lave loops på.

Den første måde er den funktionale (box 4). Her bliver defineret en function med kommandoen **(defun)** og derefter bliver den kaldt. Her bliver selve koden udført i den sidste del af funktionen. Den der starter med **(progn)**. **(progn)** giver mulighed for at udføre flere kommandoer efter hinanden.

```
Funktionelt defineret loop:
* (defun myloop (i stop)
  (if (= i stop)
      nil
      (progn
        (format t "~a~%" i)
        (myloop (1+ i) stop))))
MYLOOP
* (myloop 1 10)
1
2
3
4
5
6
7
8
9
NIL
```

BOX 5

Men der er også andre muligheder for at loope. Der findes det klassiske **for loop** som man kender fra C og andre sprog:

```

Loop-konstruktion:
* (loop for i
      from 1
      to 10
      do (format t "~a~%" i))

1
2
3
4
5
6
7
8
9
10
NIL
*

```

BOX 6

Her er det meget klassisk, med en variabel der bliver itereret fra 1 til og med 10. Man kan også iterere over en liste, som er meget brugt i blandt andet Ruby. Her tager i værdierne i listen (1 2 3 4 5 6 7 8 9 10) på skift. For hvert element udskriver den så i.

```

* (loop      for i
            in '(1 2 3 4 5 6 7 8 9 10)
            do (format t "~a~%" i))

1
2
3
4
5
6
7
8
9
10
NIL
*

```

BOX 7

Nu har vi været gennem lidt forskellige i Lisp, så nu er det tid til at lave et spil, et simpelt gæt et nummer spil, men som dog samler noget af det vi har brugt tidligere. Samt viser at det at der er en REPL gør det muligt at bruge den som input.

Først skal vi lave et lille demo program for at se om vi har logikken på plads. Vi erklærer nogle variable vi får brug for senere. Alt dette er skrevet ind i en fil, som bliver indlæst med (load).

```
(defparameter *secret-number* 5)
```

```
(defparameter *input-number* 4)
```

Nu sætter vi altid vores tal, der skal gættes til 5. Det er ikke særlig godt, så det gør vi noget ved senere.

Så til selve logikken i spillet. Vores første version bliver bare en simpel if sætning:

```
(if (= *secret-number* *input-number*)
    (format t "You guessed correctly~%")
    (format t "You guessed wrong~%"))
```

Her kigger vi bare om de er ens og meddeler brugeren dette. Vi kan lige prøve at køre det vi har indtil videre:

```

* (defparameter *secretnumber* 5)
  (defparameter *inputnumber* 4)
  *SECRETNUMBER*
  *
  *INPUTNUMBER*
  * (if (= *secretnumber**inputnumber*)
        (format t "You guessed correctly~%")
        (format t "You guessed wrong~%"))
  You guessed wrong
  NIL
  *

```

BOX 8

Den siger at vi svarede forkert. Det er OK. Vi svarede slet ikke - svaret er indtil videre hard-kodet ind i input-number, og "svaret" er jo forkert. Men lad os gøre det smartere så den kan sige om vi gætter for lavt eller højt. Til det bruger vi en (cond).

```
(cond
```

```
((< *input-number* *secret-number*) (format t "Too low~%"))
```

```
((> *input-number* *secret-number*) (format t "Too high~%"))
```

```
(t (format t "You guessed correctly~%"))))
```

Nu får vi et lidt pænere output, så lad os forsætte med (cond) løsningen.

Så lad os lave det endelige spil, med tilfældige tal og brugerinput. Det er skrevet direkte i REPL'en, fordi vi allerede har tastet variablene ind. Det kunne også skrives i en fil og loades ind, men så skal vi også først have de linier, som definerer den ene variabel secret-number, som vi bruger:

```
; box 9 - hele programmet i en fil,
; filnavn "simpelt-spil-box9.lisp"
; definition af variabel secret-number.
```

```
(defparameter *secret-number*5)
```

```
(defun newgame())
```

```
(progn
```

```
(setf *secret-number* (1+(random 10)))
```

```
()))
```

```
(defun guess (g)
```

```
(cond
```

```
((< g *secret-number*)
```

```
(format t "Too low~%"))
```

```
((> g *secret-number*)
```

```
(format t "Too high~%"))
```

```
(t
```

```
(format t "You guessed correctly~%"))))
```

BOX 9

Når man skriver Lisp programkode ind i en fil, skal man selvfølgelig ikke have stjernen med. (Stjernen, asterisk, husker vi, er SBCLisp's prompt, som fortæller at LispREPL'en er klar til mere input.)

Den første kodelinie:

```
(defparameter *secret-number* 5)
```

er definition af en variabel eller ændring af værdien af en eksisterende variabel. Stjernerne omkring navnet sikrer at det er en global eller special variabel, i LISP kaldet *dynamisk scope*.

Dernæst kommer definitionen af en funktion, som sætter vores hemmelige tal til en tilfældig værdi mellem 1 og 10, hvorefter spilleren kan begynde at gætte. Den funktion, **new-game**, sørger for at tallet, der skal gættes, er tilfældigt. Lisp funktioner returnerer altid værdien af det sidste udtryk, den har evalueret, så hvis vi ikke laver det stunt med at sætte den tomme parentes ind til sidst, vil Lisp udskrive værdien af det hemmelige tal – og så er vi jo lige vidt!

```
(defun new-game ()
  (progn
    (setf *secret-number* (1+ (random 10)))
    ()))
```

Hvis vi indtaster dette, vil Lisp kvittere med "NEWGAME" som bevis for at funktionen er godtaget. Du har sikkert undervejs fået en del kryptiske fejlmeddelelser og så må man jo gå ud af programmet (brug CTRLZ, kill %% hvis du kører Linux/Unix).

Til at lave det tilfældige tal bruges funktionen `random`, som er indbygget i Lisp. Den får besked på at vælge et tal, som er mindre end 10, medregnet nul, og da vi ikke vil have 0 med, lægger vi én til.

De sidste linier er dem, du har set før, hvor vi vurderer brugerens input og derefter skriver en pæn besked, om det var for lav, for højt eller om spilleren gættede det rigtige tal.

Referencer:

1. http://da.wikipedia.org/wiki/Alonzo_Church
2. <http://da.wikipedia.org/w/index.php?title=Lambdakalkyle>
3. http://en.wikipedia.org/wiki/Lambda_calculus
4. http://da.wikipedia.org/wiki/John_McCarthy
5. http://en.wikipedia.org/wiki/Steve_Russell
6. <http://da.wikipedia.org/wiki/Fortran>
8. <http://landoflisp.com/>
9. <http://www.gigamonkeys.com/book/>

Nu er vi klar til at spille, så lad os starte et nyt spil og gætte lidt. Hvis programmet er i en fil, vil det se sådan ud:

```
hero$ sbcl
This is SBCL 1.0.45.debian, an implementation of ANSI Common Lisp.
...
*(load "simpeltspilbox9.lisp")
T
NIL
*(new-game)
NIL
* (guess 5)
Too high
NIL
* (guess 2)
Too high
NIL
* (guess 1)
You guessed correctly
NIL
*
```

BOX 10

Først startes Lisp fra kommandolinien. Sbel svarer med versionsnummer osv.osv.osv. og til sidst stjerne, som viser at den er klar til programkode. Derefter loades programfilen (fra box 9). Husk gåseøjne omkring filnavnet og dér kvitterer Lisp med T for true (det skyldes den ekstra tomme parentes i **new-game**, og til sidst igen stjerne, den er klar til mere input.

Nu gætter vi ved at kalde vores nydefinerede funktion **guess**, som skal have et tal med, nemlig vores gæt. Programmet sammenligner og skriver svaret ud – i dette tilfælde var gættet for højt, så vi fortsætter med et lavere gæt. Bliv ved!

Vi gættede sandelig tallet. Nu kan man lege lidt med programmet, sådan som fx. at sætte tallet op så det er svære at finde det rigtige tal, eller vi kan tilføje en mekanisme til at til at tælle antal forsøg.

Læs mere:

Der findes flere steder, hvor du kan lære mere om dette fantastiske smukke sprog, men to bøger som er rigtig gode er:

Land of Lisp

<http://landoflisp.com/>

og Practical common lisp

<http://www.gigamonkeys.com/book/>

Begge to er rigtig gode ressourcer til at lære dette smukke sprog, der har rod i AI fra gammel tid.

4 NODES IN 2U

- RENDER FARM NÅR PLADSEN ER TRANG

AF: JON BENDTSEN jon.bendtsen@jonix.dk



Jon Bendtsen

Hos en af mine kunder havde de et voksende behov for en render farm som kan rendere deres 3D billeder som de laver i Daz Studio 3D. Til at starte med brugte de bare udviklernes desktop computere, hvilket gik fint til at evaluere brugbarheden af Daz Studio 3D. Men til produktion ville de bare sætte desktop maskinerne ud i serverrummet – "mit" serverrum hvor pladsen er trang. Så jeg sagde nej, og lovede at finde en bedre løsning.

Løsninger

Jeg vidste allerede at jeg ville have noget rack-mounted samt at selvom Daz Studie 3D ikke brugte grafik kortet til renderingen, så var der alligevel et mindste krav for at kunne starte applikation:

OpenGL 1.3 compatible graphics card with at least 128 MB RAM (Hardware accelerated OpenGL 2.2, or higher, compatible recommended with 256MB+ RAM)

hvilket ikke er særlig udbredt i server hardware. Yderligere ville jeg gerne minimere pladsen i mit rack, for der er kun 1 rack og det er ved at være fyldt op. Jeg ønskede også redundant strømforsyning (PSU), en til hver UPS. Jeg faldt derfor for en SuperMicro SuperServer 6026TT-HTRF fra deres Twin2 serie, der indeholder 4 servere på kun 2U. De 4 servere (fremover kaldt node) er fordelt på 2 i hver 1U og 2 i hver side, altså en 2x2 matrice. Hver node ligger i sin egen skuffe som kan trækkes ind og ud helt uafhængigt af de andre noder, som bare kører videre¹.

Hver node fylder lige lidt under 1U i højden og en halv i bredden. Det gør at der kun er meget lidt plads til et grafik kort som var nødvendigt. Manualen

fortalte ikke mere præcist end "halv højde". Jeg skrev og spurgte deres support, men de sagde det samme. Jeg måtte have distributøren til at undersøge og sikre at vi kunne putte et passende grafik kort i maskinen. Kortet skulle selvfølgelig ligge ned, så højden bliver vel til bredde? Foruden det var der restriktioner på tykkelsen samt på længden hvis vi ønskede at bruge 2 CPU'er. Da Daz 3D rendering ikke altid kan udnytte flere cores, nøjes vi med 1 low-end xeon 4 cores i stedet for 2 cpuer, derfor var længde restriktionen ikke aktuel.

Distributøren var dejlig grundig og chekkede også for evt. varme og strøm restriktioner. Heldigvis kom de tilbage og svarede at man godt kunne putte 1 passende grafik kort i hver node.

Grafik kortet er et TUL VTX HD5450 512MB DDR3 PCI-E.

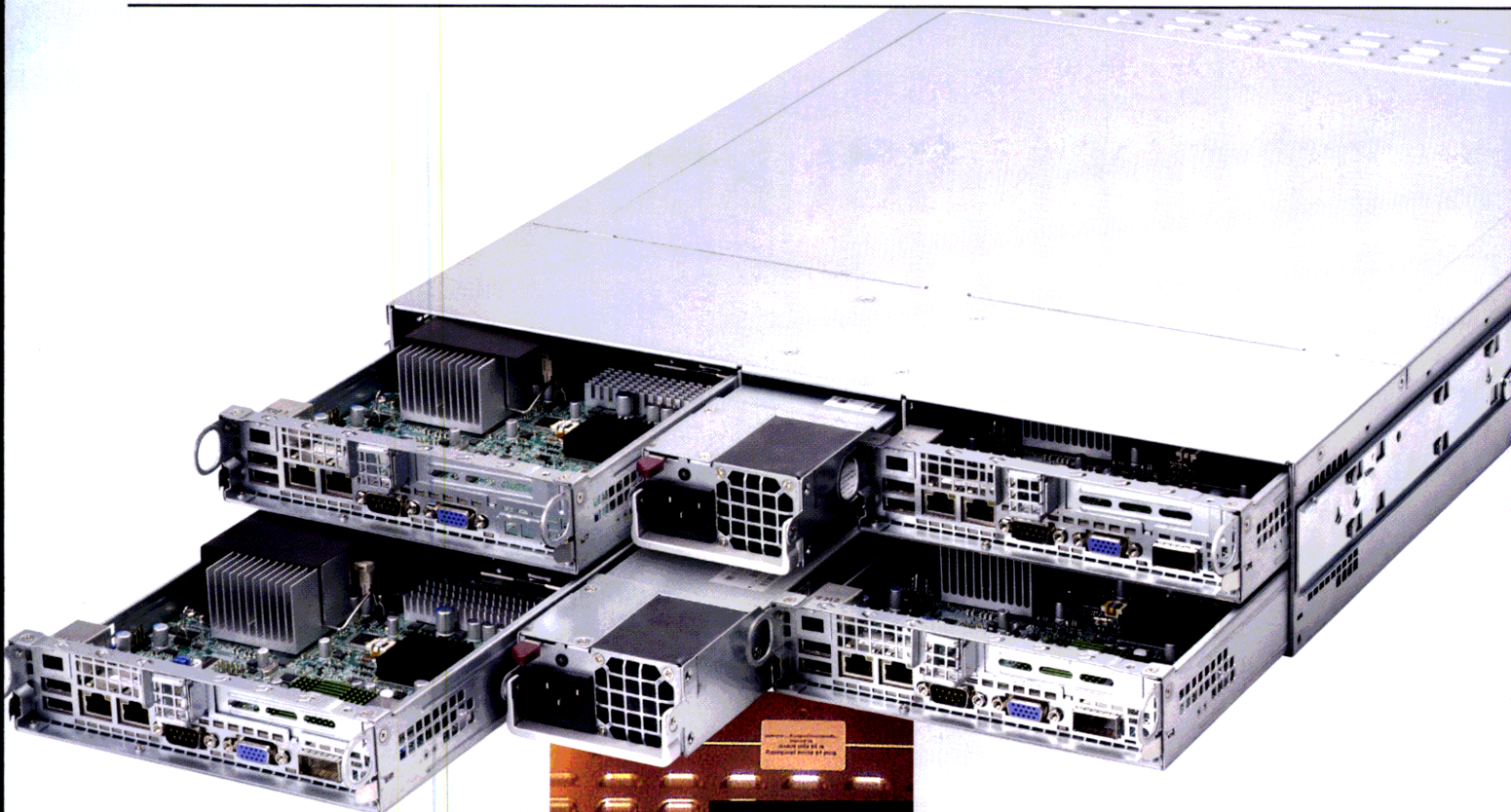
Efter jeg fik maskinen i hånden så skyndte jeg mig at måle hvor meget plads der er til indstikskort, hvis andre har behov for oplysningerne.

max 175 mm langt
max 70 mm højt målt fra toppen af PCI soklen.
under 20 mm tykt

Hvis der ikke havde været plads, og jeg stadig gerne ville have haft redundant PSU (hvilket er svært at få i 1U maskiner), så kunne jeg have købt en 2U server med 2x 2U's noder fra SuperMicros 2U Twin serie. Hver af disse noder kan have lidt flere PCI kort.

Bundkortene som passer i node-skufferne fås desværre ikke med onboard 10Gbit ethernet, men al-

¹) Salgsmateriale, ikke afprøvet af Jon Bendtsen i praksis.



ternativt kan man få dem med Infini-Band² eller selv putte et 10Gbit PCI kort i dem.

Diskskuffer

Min server kommer med 12x 3,5" skuffer til SATA diske, 3 til hver node, men jeg har kun installeret 1 almindelig desktop harddisk i hver node, for applikationen er slet ikke disk begrænset. Går der en disk, så har vi et image til hurtig restore.

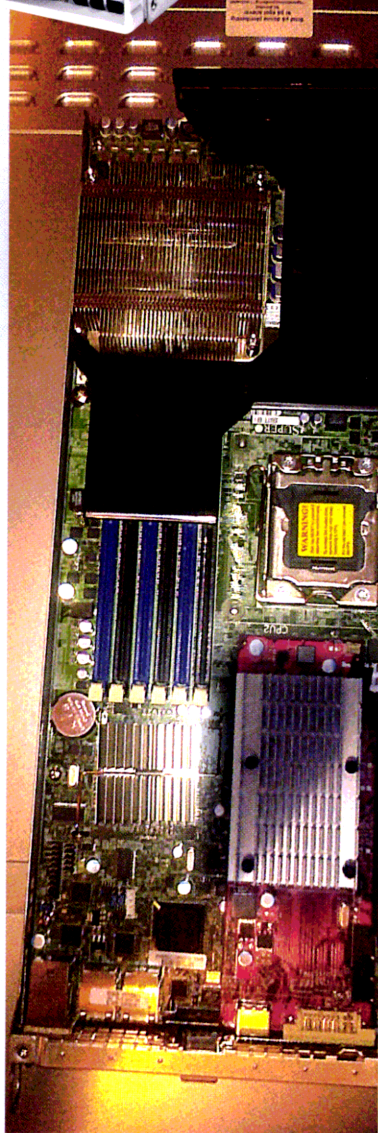
Hvis du i stedet ønsker 2,5" diske, fx til SSD diske, så kan man købe et andet kabinet som kommer med 24x 2,5" skuffer, 6 til hver node.

Den larmer

Denne server larmer rigtig meget, specielt lige når man starter den. Lyden mindede mig om opstarten på de civilforsvarsalarmsystemer som hylede hver onsdag klokken 12 da jeg var knægt.

Kørt upåklageligt

Serveren, eller skal man sige serverne for der er jo 4 maskiner? har kørt upåklageligt i små 2 måneder og har



givet den forventede renderingskraft i forhold til hvad brugerne kendte fra deres i7 desktop maskiner.

Udviklerne har nu også spurgt hvad det ville koste at fordoble performance, og her er svaret jo ganske simpelt, køb en tilsvarende løsning, for det bliver hurtigt meget dyrt, 7k pr. node, at putte en 3GHz Xeon CPU i. Så nu har jeg 2 SuperMicro SuperServer 6026TT-HTRF servere.

Alternativ til et blade center.

Denne SuperMicro server er et ganske glimrende alternativ til en bladecenter løsning hvis man ikke har behov for særlig mange fysiske servere, eller vil starte for små penge.

Læs mere:

<http://www.daz3d.com/>

<http://supermicro.com/products/system/2U/6026/SYS-6026TT-HIBQRF.cfm>

²⁾ <http://en.wikipedia.org/wiki/Infiniband>

OPEN SOURCE DAYS 2012

Have an overview over planned talks so far.

WEBKIT, WHY IT MATTERS

WebKit is the engine behind most mobile browsers and powering Apple's Safari and Google's Chrome. But how did it all start? Are there other places where WebKit is in use? How does such a big open project work?

And what's up with all the different ports? Have you at some point asked yourself one of these questions or are you just interested in getting to know more about this highly successful open source project, then this is a talk you do not want to miss.

The speaker has a long-time involvement with the project and will be able to answer the above questions. Other topics will also be touched such as having a look at where the project and HTML5 is heading, and how to contribute to the project or just how to use it yourself in your own projects.

SPEAKER: KENNETH CHRISTIANSEN

Kenneth Christiansen is a Senior Engineer at Nokia Mobile Phones, where he is focusing on bringing HTML5 and related technologies to mobile devices, while still keeping the user experience a top priority. Beyond his interest in exploring new ideas for developing the next generation browser, Kenneth stays up to date on the latest web technologies as additionally a team member

of the distributed Qt WebKit team, and as an acknowledged WebKit reviewer. Kenneth holds a MSc degree in Computer Science in the field of Software Engineering/Computational Science and Visualization.

Subjects: Webkit, Open Source project, HTTP5.

MODERN WEB PROGRAMMING IN PERL

In year 1997 dynamic web pages often consisted of HTML forms you could submit to the server and get a full HTML page back. The standard language for writing the server code was Perl and it is quite probable that the worst code ever written in any language is written in Perl about 15 years ago to handle some HTML form processing.

Since then different languages have been hyped as *the* language for server side web development: PHP, Java Server Pages, Ruby on Rails, and latest JavaScript in the form of node.js.

But Perl isn't the same as it was 15 years ago. The community has matured, developed new ways of using Perl, and even acquired ideas from other language communities. During this talk I will show the modern way of developing web services in Perl. Speaker: Peter Makhholm About the speaker: Peter

Makhholm has been programming in Perl for 15 years, but only been developing for the web for 10 years, mostly in Perl and PHP.

Today he develops deep backend stuff for One.com, often (ab)using the HTTP protocol.

SPEAKER: PETER MAKHOLM

Peter Makhholm has been programming in Perl for 15 years, but only been developing for the web for 10 years, mostly in Perl and PHP. Today he develops deep backend stuff for One.com, often (ab)using the HTTP protocol.

Subjects: Web programming, Perl.

MASS DEPLOYMENT OF UNIX MACHINES

There is no doubt that open source unices have proven themselves capable of the efficiency and stability required from modern enterprise operating systems. The number of large scale unix installations grow month by month, and once an installation grows larger than a few dozen machines, it quickly becomes infeasible to deploy them one by one using physical installation media.

Luckily, memory gets cheaper and cheaper, and storage virtualisation isn't nearly as hard as it was just 5

years ago. Thus, one way of easing mass deployment is to separate the system installation from the hardware entirely.

This talk will demonstrate a simple proof of concept of how to do scalable mass deployment of diskless unix nodes, using simple techniques and open standards.

SPEAKER: JESPER NYERUP

Jesper Nyerup is a Unix systems administrator with a larger Danish hosting provider. He mainly keeps himself busy doing hardware integration, datacenter management and deployment procedures. Linux user since 2.0.34, but still kernel panicking several times a month.

Subjects: Unix.

OCCUPY WALL STREET'S INFORMATION TECHNOLOGY: A FACEBOOK FOR THE 99%?

The Occupy Wall Street (OWS) Movement has always used open-source technologies to reach the wider public and to coordinate internal activities. However, with the eviction of the overwhelming majority of encampments, particularly the Wall Street Encampment, the online presence has become crucial to the survival of the Movement.

We will briefly review both practical and theoretical issues that have been encountered during software development for the Movement. One practical problem is to identify appropriate projects. Another problem is to assemble

developers, designers, project managers, and others for a given project. Finally, how can these people then maintain motivation and coordinate their activities?

One major project, dubbed "Facebook for the 99%" by Wired Magazine, is now through the stage of User Experience investigation. Information Architecture design, however, has presented some difficult decisions. Some argue that the well known triple, User-Role-Scope, can be extended in order to meet the needs of the Movement. Others maintain that a break needs to be made with the Facebook and similar approaches to groups as merely a collection of users. This argument sees Groups as greater than the sum of their parts and wishes to have Groups as first-class objects in the new Information Architecture. Other discussions are concerned with how to connect with similar movements worldwide, such as Take the Square, which has been using N-1, a distributed social networking system. The underlying question is whether the Movement just needs to do new things with available open-source technologies or whether new technologies have to be developed to meet the needs of the Movement.

SPEAKER: DAVID STODOLSKY

David Stodolsky (BA, University of Michigan, Ann Arbor; MA University of Wisconsin, Madison; PhD, University of California, Irvine) did post-doctoral research at the University of California, Berkeley and at Stanford University in social-psychology. The research focused upon group problem solving when computer executable rules were used to resolve conflicting requests to speak in teleconferencing environments. Current research interests include security mechanisms for enhancing

personal integrity in communication and health care systems, management of argumentation and commitments in computer conferencing systems (most recently with an emphasis upon peer review mechanisms), and real-time group decision support.

Subjects: Software for collaboration, Open Source.

AN UPDATE ON OPEN SOURCE BUSINESS MODELS AND LEGAL ISSUES

In his talk, Martin von Haller Groenbaek will provide an update on the latest news regarding open source license, legal development and the importance for open source business models, such as the adoption Mozilla Public Licence Version 2.0, Open Source in the Cloud, the Patent Wars and other things.

SPEAKER: MARTIN VON HALLER GROENBAEK

Martin von Haller Groenbaek is the leading Danish expert on legal and strategic aspects of open source. He is a partner of BvHD law firm in Copenhagen and a co-founder and currently chairman of the board of the Danish Open Source Vendors Association (OSL). He also an initial member of the Danish Creative Commons chapter. As a lawyer he has advised large and small companies as well as public and government institutions on.

Subjects: Licensing business models, cloud, patents.



QT PROJECT AND QT 5

The Qt project was established in October 2011 to drive further development of Qt as an open source project. The main development effort of qt-project.org goes into Qt 5, the next major version of Qt that is currently being developed. The talk will explain the setup of qt-project.org, and how people can participate and contribute to the project. It'll also give an overview over Qt 5.

Topics:

- Open Source Community
- Mobile Platforms
- Programming

Abstract:

The Qt toolkit has existed since the mid-90s and has been used for development of many desktop and embedded applications, including the entire KDE desktop and tools such as VLC. It has found considerable success in the embedded segment, being used to develop the UI and UX of many devices, ranging from medical to automotive, industrial all the way to domestic appliances like coffee makers and refrigerators.

In the effort to become more and more open and to attract more developers in the process, in 2011 the Qt Project was established to drive the future development of the project. A true Open Source project was born in October with the launch of qt-project.org. An open governance model was established, based on the models found in mature open source communities like the Linux kernel and WebKit, with very

easy entry path, a simple process to get contributions in and with a group of maintainers responsible for the overall quality and direction.

The main goal of the Qt Project for 2012 is to launch Qt 5.0, the first major version of Qt in 7 years. Following the goals outlined last May, Qt 5.0 contains as its most important feature an overhaul of the graphics architecture, preparing it for the next 5 years or more of UX evolution.

The talk will focus on how the Qt Project's community is set up, the interactions between the community members and show how a new-comer can approach the project with their ideas and contributions. The last part of the session will show what the most important features of Qt 5 are.

SPEAKER: THIAGO MACIEIRA

The presenter is the maintainer of the QtCore and QtDBus libraries and was the lead person for the creation of the Qt Project.

Subjects: QT5 project, Open Source, mobile platforms, programming.

INTERESTING QT5 ADDONS AND QT5 BASED SOLUTIONS

A brief walkthrough of some of the interesting Qt Addons and Qt based Solutions that KDE is planning to release for Qt5 shortly after the initial release of Qt 5.0. This includes libraries for doing asynchronous API's, IMAP library, hardware abstraction API's and larger modules for file indexing and file search.

SPEAKER: SUNE VUORELA

Sune Vuorela is a long term Debian and KDE developer and has recently gotten patches into Qt. Sune also has several years experiences in doing Qt apps professionally.

Subjects: QT, KDE.

WHY JOOMLA?

Why Joomla? A unique open source development and governance model
Description: Joomla! has no corporate backer, no paid employees, no membership and no benevolent data so how it does this all work and what are the challenges. Is this a model that could work for you?

SPEAKER: BRIAN TEEMAN

As co-founder of Joomla Brian Teeman spend a considerable ammount of his time travelling the world and presenting to large audiences about Joomla and he is also a professional trainer.

Subjects: Joomla, Open Source, CMS.

GITIFY YOUR LIFE

Git is a distributed version control system which is used to manage the source code of the Linux Kernel, KDE, and more.

While git was written to manage source code, it can do a lot more!

From tracking personal notes to managing your website, wiki, and blog over tracking system and personal configuration files to managing videos, photos and other large files and making system backups, a lot of tools have been grown around the git ecosystem to help you support most tasks of your digital life.

This talk will show you a lot of neat tools and tricks and it's highly likely that you will adopt at least one of the various solutions.

The talk will end by showing a tool to manage your myriad of newly created git repositories; its configuration can, you guessed it, be maintained by git as well.

SPEAKER: RICHARD HARTMANN

Richard "RichiH" Hartmann is a member of the board of the PDPC, freenode staff, and OFTC staff. His dayjob being a project and backbone management at Globalways AG, many of his nights are spent around FLOSS in general. His main interests hover around command line tools, data integrity, FLOSS communities, cold countries, and scuba diving.
Track:

Subjects: Git usage, creativity, organizing your life, planning.

NOSQL

An introduction of the most common algorithms used in popular NoSQL databases; how does the math work, how it is implemented, the tradeoffs, and how it affect the databases build using them. If either stuff like vector clocks, paxos, consistent hashing, CAP rings a bell, or some of the products built using them, BigTable, Cassandra, HBase, Riak, MongoDB, Hadoop, piques your interest, this should be informative.

Requires basic knowledge of distributed computing and database concepts.

Speaker: Morten Siebuhr

Morten Siebuhr is a Computer Science master from University of Copenhagen, with a particular interest for distributed systems. Currently works as a developer for One.com.

Subjects: Databases without SQL, data organization, algorithms.

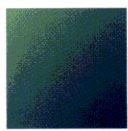
ARDUINO PROTOTYPING PLATFORM: FROM COMMAND LINE TO BLOCK PROGRAMMING FOR BEGINNERS IN ELECTRONICS

This talk will go through the history of the Arduino platform. We will see the latest developments as well as different lines of development where to join the community and contribute to making it a better development environment for everyone interested in embedded electronics, robotics, and the internet of things.

SPEAKER: DAVID CUARTIELLES

David Cuartielles is Co-founder of the Arduino platform. Holds a Research Fellow position at Malmo University and is a PhD candidate in Interaction Design.

Subjects: Arduino programming, prototyping.



OPEN SOURCE DAYS 2012

Kurser

Som en nyhed er der i år kurser på Open Source Days, som afholdes på dagene inden selve konferencen:

TORS DAG D. 8. MARTS.

FREDAG D. 9. MARTS.

Kurseemnerne spænder vidt, fra introduktion og moderne webprogrammering med javascript libraries node.js og jQuery - over PostgreSQL og netværksopbygning med OpenBSD pakkefilter - til software lifecycle management med Git, Gerrit & Jenkins.

Kurserne afholdes på engelsk.

INTRODUCTION TO JAVASCRIPT FOR MODERN SOFTWARE DEVELOPMENT

Javascript vokser i popularitet, og ikke kun for webudvikling. Et stigende antal implementationer findes, og 3.part biblioteker er under udvikling. I dag anvendes Javascript i komplekse front-end/ browser løsninger såvel som på servere og til mobil løsninger. Dette kursus vil introducere sproget fra bunden og desuden vise eksempler på anvendelse af 3. parts biblioteker, såsom jQuery og Node.js.

Kenneth Geisshirt er kemiker af uddannelse, og nørd af natur. Han har

brugt fri software siden begyndelsen af 1990'erne. Han arbejder som freelance software udvikler og tech-forfatter.

Tid: Torsdag d. 8. marts

Start: 8:30,

Slut: 17:00.

Minimum/max deltagere: 8/20

DESIGN PATTERNS

Design Patterns eksploderede indenfor software udvikling omkring 1994, da Erich Gamma, Richard Helm, Ralph Johnson og John Vlissides publicerede bogen "Design Patterns: Elements of Reusable Object-Oriented Software". Forfatterne er sidenhen kendt som "Firebanden", Gang Of Four, (GOF). Succesen var en overraskelse for forfatterne, som "bare" ønskede at dokumentere kendte løsninger til almindelige software problemer.

De havde set de samme problemer løst på forskellige måder igen og igen, og syntes at det var spild af udviklernes tid. Denne bog er nu kommet i 36'te oplag.

Denne trænings-session vil gennemgå mange Design Patterns fra bogen, og deltagerne vil arbejde med praktiske kode-eksempler i løbet af dagen.

Design patterns er baseret på Objekt-Orientering, og bruges som grundlag når man arbejder med kode-refactoring.

Java/Eclipse vil blive brugt til alle eksempler og øvelser, så deltagerne bør medtage en laptop hvorpå der er installeret Eclipse.

Sverre Eplov har arbejdet som software udvikler, arkitekt, kursusinstruktør og projekt-manager siden 1992. Sverre har hjulpet med at introducere Design Patterns og Refactoring i flere større organisationer.

GET STARTED WITH PYTHON DEVELOPMENT

Sigtet med kurset er at komme i gang med at udvikle scripts i Python. Efter kurset vil du være i stand til at skrive scripts, som læser og skriver filer, forstå kommandolinie-switches og -parametre, bruge regulære udtryk (regular expressions) og levere passende dokumentation.

Forudsætninger: Generelle computerfærdigheder, herunder at kunne bruge en teksteditor, fortrolighed med kommandolinie-programmer, og tekstmanipulation.

Overskrifter:

- Python, hvad er det og hvad kan jeg bruge det til?
- Værktøjerne
- Strukturen af et pythonscript

- Tal og tekststreng
- Kontrolstrukturer
- Funktioner
- At skrive dokumentationstekststreng
- Lister, tupler og dictionaries
- Exceptions og fejlhåndtering
- Kommandolinie-parametre og environment-variable
- Regular expressions

Hvordan gør jeg mig klar til kurset?

Installer Python 2.7 (2.6 og 2.5 er også OK), IDLE udviklingsmiljøet, og hav din favorit teksteditor parat i skudhøjde.

Thomas Ammitzbøll-Bach er en old-timer i Linux-miljøet i Danmark. Han var en af grundlæggerne af SSLUG i 1996, og har ofte været taler ved Linux Forum og Open Source Days. Thomas har programmeret siden 1980, brugt Linux siden 1994, og har skrevet Python siden 2005. Thomas er indehaver UU.DK, et firma, som tilbyder kurser og konsulent-tjenester indenfor et bredt spektrum af open source teknologier og softwareudvikling.

BUILDING THE NETWORK YOU NEED WITH PF, THE OPENBSD PACKET FILTER

Dette endags-kursus sigter mod erfarne såvel som kommende netværks-administratorer, som ønsker at udvide deres kendskab til PF (Pakke-Filter), og de relaterede værktøjer. En grundlæggende viden om Unix og TCP/IP netværk konfiguration forventes og er nødvendig. Emner, som der arbejdes med, omfatter:

- Konfiguration på OpenBSD, FreeBSD og NetBSD

- PF grundlæggende regelsæt og interaktion mellem regler: Blokér, lad passere, match
- Om at skrive regelsæt, som er lette at vedligeholde
- Adresse-familier: IPv4 NAT vs. IPv6
- Redirigeringer, servicer med mærkelige afhængigheder, (ftp-proxy, spamd)
- Adaptive regelsæt (tips vedrørende status tracking)
- ALTO traffic shaping (båndbredde pr. trafiktype)
- Filtrering pr. bruger med authpf
- High availability med CARP, relay
- Trådløse versus trådede netværker
- Filtrerings-broer
- Logning og overvågning - pflog, pflow og andre
- Test, fejlfinding og optimering af jeres konfiguration

Dette kursus vil give opdateringer vedrørende den nye PF syntaks og features, som blev introduceret med OpenBSD 4.7 (med eksempler præsenteret i gammel og ny syntaks, hvor det er relevant), og desuden med nyere opdateringer og smuglig på relevante nye features i den kommende OpenBSD 5.1 (planlagt release 1. maj 2012).

Kursusinstruktør: Peter N. M. Hansteen

MODERN MAIL MANAGEMENT

Et velgennemtænkt setup kan gøre afgørende forskel når det drejer sig om at køre et automatiseret mailsystem, som ikke behøver meget indgriben fra postmesteren.

I dette en-dages-kursus om "Moderne Mail Management" vil vi tage et nærmere kig på de betingelser, som et mail system må kunne klare i dag. Vi vil diskutere mulige implikationer

og udvikle retningslinier, som skal tage højde for de nødvendige ting, og andre ting, som vi ønsker at håndtere.

Når vi er blevet enige om en plan ruger vi Open Source bedste-praksis værktøjerne, Postfix, ClamAV, SpamAssassin, amavisd og Dovecot til at bygge et mailsystem, som kan følgende:

- filtrere indkommende spam, vira og andre specielle mail-klasser, - anvende forskellige policies for indkommende og udgående trafik, - forbedre afsender domænets evne til at komme igennem til modtagerne,
- hjælpe interne sendere med at skabe formelt set korrekte mail,
- kan opdage udgående spam.

Vi vil slutte dagen med et peer review, hvor vi tager et kig på trainee produktions-opsætninger. Hvis tiden tillader det, og hvis det er muligt, vil vi diskutere og derefter implementere forbedringer.

Bio: Ralf Hildebrandt arbejder ved Charite hospitalet i Berlin. Han er co-author af bogen "The Book of Postfix", "Postfix - Einrichtung, Betrieb und Wartung", "Linux Email" og forskellige artikler.

Patrick Ben Koetter er Master of American Studies ved Ludwig-Maximilians Universität i München, men det afholdt ham ikke fra at lodde sin fars gramofon i en alder af 11 år. Det var da hans far besluttede at kanalisere Patrick's konstante krav om at lege med teknik og begyndte at gå til klasser i det lokale kursuscenter.

Læreren var en fyr, som kørte et firma der fremstillede computere. Patrick brugte resten af sommeren på hans fabrik, exponerede printkort til UV lys og købte sig selv et stereo anlæg (inklusive en gramofon til sig selv) ved afslutningen af dette praktik- ▶



ophold. I dag er han professionel IT konsulent og kursusinstruktør, som kører sit eget firma.

Indenfor sit team har Patrick specialiseret sig i Informations-Arkitektur, brugergrænseflader og E-post. Sammen med Ralf Hildebrandt har Patrick skrevet "The Book of Postfix", "Postfix - Einrichtung, Betrieb und Wartung", og han skriver også regemæssigt artikler for blade.

Hvis tiden tillader det, prøver Patrick at forbedre open source software dokumentation. Patrick har skrevet Amavis' officielle Postfix Howto. Alt i alt er Patricks evner forøget betragteligt gennem tiden - evnen til at skabe ting og også ødelægge dem ... ;-)

Instruktør: Ralf Hildebrandt & Patrick Ben Koetter

Emner: Postfix ClamAV SpamAssassin amavisd Dovecot

Tid: Fredag 9. marts 2012

Start: 8:30.

REFACTORING

Med udgangspunkt i Martin Fowlers bog, "Refactoring: Improving the Design of Existing Code" vil vi på dette kursus gennemgå hvad refactoring faktisk er, og hvordan man bruger det i et praktisk miljø.

Refactoring er en disciplin fyldt med faldgruber, men brugt korrekt kan den gøre det muligt for udviklerne at arbejde med strukturen af kodebasen, mens man holder kodebasen strukturelt sund.

Deltagerne vil få en introduktion til hvordan man bruger Martin Fowlers

bog, og nogle praktiske tips til hvilke tegn man skal holde øje med i koden, og hvile forholdsregler der kan tages for at undgå kode-forrådelse.

Der kommer praktiske eksempler og øvelser i Java, så deltagerne bør medbringe en laptop med Eclipse installeret.

Bio: Sverre Eplöv har arbejdet som software udvikler, arkitekt, instruktør, og project manager siden 1992. Sverre har bidraget med at introducere Design Patterns og Refactoring teknikken i adskillige større organisationer.

Instruktør: Sverre Eplöv

Emne: Design software decay software rot

Tid: Fredag 9. marts 2012

Start: 08:30

Slut: 16:30

Status: Open for registration Buy ticket

Minimum/max attendees: 5/10

Sted: Symbion, Fruebjergvej 3, DK-2100 København Ø

POSTGRESQL REPLICATION

Der er ikke nogen magisk løsning for al slags behov for replikering og high availability, og det er ofte effektivt at kombinere flere metoder til én løsning. Dette kursus begynder med et overblik over de mange forskellige availability løsninger, som findes for PostgreSQL, deres fordele og ulemper, og typiske scenarier for anvendelse.

Anden del af kurset vil specifikt dække Streaming Replication introduceret i PostgreSQL 9.0 og forbedret i 9.1, inklusiv transaktions kontrolleret synkron replikering.

Tidligere erfaring med PostgreSQL er ikke et krav, men vil formentlig sikre et større udbytte af kurset.

Bio: Magnus Hagander er medlem af PostgreSQL core team og er udvikler og kode-committer i PostgreSQL Global Udviklingsgruppe.

Magnus er en af de oprindelige udviklere af Windows-port af PostgreSQL og aktuelt er han del af det hold, som vedligeholder den. For tiden arbejder han på andre dele af PostgreSQL backend, i den seneste tid med fokus på sikkerheds features, monitorering og backup/replikering interfaces og værktøjsprogrammer.

Han er også kernemedlem af postgresql.org infrastruktur mandskabet, som vedligeholder de servere, der er basis for projektet, og en administratøren på postgresql.org websitet. Han bidrager til pgAdmin og andre relaterede projekter.

Han har brugt PostgreSQL siden version 6 (med noget mindre seriøs brug af PostgreSQL 95 før den tid) og i øjeblikket er han med i Core Team og som præsident for The Board for PostgreSQL Europe.

For at kunne betale regninger er han PostgreSQL og open source software konsulent i Redpill Linpro i Stockholm, hvor han arbejder med rådgivning, vedligehold og kursus-tjenester, såvel som kundespecifik programudvikling.

Instruktør: Magnus Hagander

Emne: sql databases

Tid: Fredag 9. marts 2012

Start: 08:30

Slut: 16:30

Status: Open for registration Buy ticket

Minimum/max attendees: 5/15

Sted: Symbion, Fruebjergvej 3, DK-2100 København Ø

DNSSEC AND OPENDNSSEC

Overskrifter:

- Kort introduktion til DNSSEC og OpenDNSSEC applikationen
- Forudsætninger for at køre OpenDNSSEC og beskrivelse af lab-opsætning
- Praktisk planlægning af jeres DNSSEC policies, baseret på tre reference cases.
- SoftHSM installation og initialisering
- Konfigurationsfiler for ODS, conf.xml, kasp.xml og zonelist.xml
- Filer og foldere som bruges af ODS
- Drift af OpenDNSSEC
 - Signering af en enkelt zone
 - Listning af keys
 - Export af public keys
 - Fremstilling af en ZSK, manuelt og automatisk
 - Fremstilling af en KSK
 - Signering af mange zoner
- Monitorering af systemet med Audit programmet
- System administration
 - Backup
 - Key administration
 - Logning

Patrik har 15 års erfaring som professionel bag sig, og har været aktiv som konsulent siden 1996. Før det arbejdede han med system administration og undervisning på Chalmers Teknologiske Universitet i Göteborg. Patrik har stor viden indenfor software udvikling, system- og protokoldesign. Viden og erfaring, som han anvender til forskning og udviklingsprojekter så vel som til at udføre dybde analyser af sikkerhed og kritiske systemkomponenter.

Aktuelle opgaver omfatter ekspert rådgivning vedrørende netværk og IT

sikkerhed, i Sverige og internationalt. Blandt Patriks mere specifikke ekspertise-felter er DNSSEC, VoIP, IPv6 og anvendt kryptografi, og han er en efterspurgt foredragsholder og kursusinstruktør indenfor disse områder.

Patrik er anerkendt indenfor organisationer som Internet Engineering Task Force (IETF) and Réseau IP Européens (RIPE). Han er forfatter til adskillige Internet Standards og specifikationer, (IETF RFC). Han er også én af de aktive udviklere bag OpenBSD, OpenSSH og OpenDNSSEC.

Instruktør: Patrik Wallstrom

Emne: DNSSEC OpenDNSSEC security administration

Tid: Fredag 9. marts 2012

Start: 08:30

Slut: 16:30

Status: Open for registration Buy ticket Minimum/max attendees: 10/20

Sted: Symbion, Fruebjergvej 3, DK-2100 København Ø

APPLICATION LIFECYCLE MANAGEMENT WITH GIT, GERRIT & JENKINS

Management og forståelse af /End to End/ (E2E) flow af højkvalitets-software fra udvikling til deployment er en nøgleparameter i moderne software udvikling. Efterhånden som kodebaser og antallet af udviklere vokser er det bydende nødvendigt at sikre sig eventuelt gennemtvunge kvalitetscheck gennem hele processen. Idag kan dette opnås helt og holdent med open source programmer, ved at bruge Git for

versionskontrol, Gerrit for kode review og branch management og Jenkins for kontinuert integration.

I dette træningsforløb vil vi dække setup og brug af Git, Gerrit og Jenkins med en kombination af præsentationer, hands-on øvelser og 1:1 coaching. Kursusdeltagere bør have noget grundlæggende Git-viden men det er dog ikke et krav. Kursister bør medbringe en laptop, hvorpå de kan installere software; Mac, Linux og Microsoft Windows kan alle bruges.

Bio: Knud Poulsen, Lars Pedersen, Emanuele Zattin and Aske Olsson

arbejder for tiden alle med Open Source programmer til Application Lifecycle Management for Nokia Mobile Phones.

Instruktør: Knud Poulsen, Lars Pedersen, Emanuele Zattin and Aske Olsson

Emne: E2E Git Gerrit Jenkins

Tid: Fredag 9. marts 2012

Start: 08:30

Slut: 16:30

Status: Open for registration Buy ticket Minimum/max attendees: 5/20

Sted: Symbion, Fruebjergvej 3, DK-2100 København Ø

Bemærk: Kurserne afholdes i Symbion Fruebjergvej 3 2100 København Ø

Priser:

Early bird: DKK 2.500,- incl. moms.

Later: DKK 3.125,- incl. moms.

Tidsplan for de fleste kurser:

08:30 - Ankomst og let morgenmad.

09.00 - Kursus start.

12.00: - Frokost

Afternoon: tea/coffee and cake

16.30 - Kursus slutter.

Bemærk: Kurserne afholdes et andet sted end selve OSD konferencen.



LISP

- ET PROGRAMMERINGSSPROG

Programmeringssprog er baseret på et væld af principper, som alle går ud på at skabe en kontrolmekanisme, som gør det nemt at arbejde med maskinen.

Eller er det ikke helt rigtigt? Det er vist lidt af en smagssag, om man anser LISP for at være nemt arbejde med. Især de mange parenteser bliver der gjort tykt grin med - også af Lisp programmører, som for eksempel forfatteren af "Land of Lisp", Conrad Barski, som har megen selvironi:



Navnet LISP er afledt af "LIST Processing" og det skyldes, at linkede lister er en af LISP sprogets vigtigste datastrukturer. Lisp Program kode udgøres i sig selv af lister. Oversætteren (compileren) bruges til at bygge Lisp. Der er dog 80 filer med C source, som er nødvendige for at bygge compileren, men det er ikke anderledes end det er med C sprogets IO-library, som har brug for nogle kilobyte maskinkode (assembly language source til IO og andre hardware-operationer).

I dette nummer fortæller David Fotel om Lisp for begyndere, og han bruger Steel Bank Common Lisp sbcl som grundlag. SBCL er Open Source og kan hentes på www.sbcl.org

Vi anbefaler sbcl fordi det er et levende projekt, bemærk dato for sidste version:

www.sbcl.org

Steel Bank Common Lisp

About

Steel Bank Common Lisp (SBCL) is a high performance Common Lisp compiler. It is open source / free software, with a permissive license. In addition to the compiler and runtime system for ANSI Common Lisp, it provides an interactive environment including a debugger, a statistical profiler, a code coverage tool, and many other extensions.

SBCL runs on a number of POSIX platforms, and experimentally on Windows. See the [download](#) page for supported platforms, and [getting started](#) guide for additional help.

The most recent version is [SBCL 1.0.55](#), released January 09, 2012 ([release notes](#)). In addition to the official SBCL, [a Windows fork](#) exists that adds support for three

non Lisp

development

Download

GCL - an implementation of Common Lisp

NEW! (20050810) GCL 2.6.7 is released. The release notes are available at [http://www.gcl.lispworks.com/NEWS/20050810.html](#)

NEW! (20050427) Patch added to the errata page to reenable the [GCL 2.6.7](#) download link.

Sidste version Gnu Lisp er fra 2005, det kan betyde at den er stabil, eller at der ikke er nogen, som interesserer sig for den.

Lisp kildetekst kan oversættes til "native bytecode", ligesom Awk og Perl og Python, og kan linkes sammen med Common Lisp environment til en binary eksekverbar fil, der ikke er voldsomt stor.

Lisp udvikling er ofte interaktiv, som i Davids artikel. Funktions og klasse-definitioner er normalt compileret og linket ind i Lisp miljøet, så snart definitionerne er skrevet. Variable kan opdateres og funktioner kan kaldes med det samme fra REPL (kommando- eller fortolker prompt).

Et Lisp program startes ved at man kalder en funktion - enten ved at indlæse kildekode eller ved anden extern mekanisme (fx. crontab etc.) Det er ikke noget, der er specielt for Lisp, sådan er det for alle sprog, der skal være et eller andet externt, som "skubber tingene igang" - det gælder også objekt-orienteret kode og C sproget, hvor det er en linker-konvention at /main()/ funktionen bliver kaldt fra operativsystemets exec+opsætning.

Lisp sproget er en klassiker. I en tid, hvor udviklingen går i retning af platformspecifikke sprog med faciliteter til applikations-building, er det interessant at notere sig, at funktionsprogrammering har fået en lille renaissance.

Fx. er der i C# v.3.0 med LINQ (Language Integrated Query) support for nogle funktionale konstruktioner.

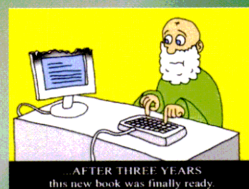
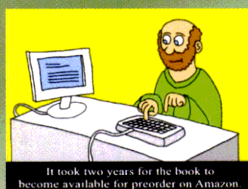
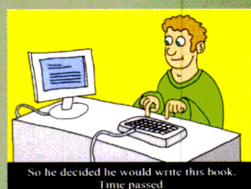
Lisp er ikke en nyhed, men det er vores opgave her i "Danish United Unix Gurues" en gang imellem at præsentere basisviden for at vi bedre kan forstå udviklingen i andre sprog. Vi skal kvalificeret følge med i hvordan det går de klassiske sprog og de grundlæggende programmeringssprog.

Se også:

http://en.wikipedia.org/wiki/Functional_programming

http://en.wikipedia.org/wiki/Lisp_%28programming_language%29

Bog og website Land of Lisp, som David anbefaler, er fuldt af sjove og lærerige instruktioner - og Land of Lisp har sin egen video, som ikke er uden selvironi!





GENERALFORSAMLING

I DKUUG DEN 28.11.2011

AFHOLDT I SYMBION – REFERAT

1. Valg af dirigent (bestyrelsen foreslår Kenneth Geissshirt). Kenneth blev valgt.

2. Valg af referent (bestyrelsen foreslår Flemming Jacobsen). Flemming blev valgt. Der blev foreslået og foretaget en kort navnerunde.

3. Beslutning om mødet er lovligt indkaldt
Keld mente ikke at regnskabet var rettidigt udsendt, idet det ikke var tilgængeligt før d. 15 kl 9. Den kommende bestyrelse bedes stramme op. Men ellers mener generalforsamlingen ikke at det er et problem.

4. Fremlæggelse af formandens og udvalgenes beretninger.
Udvalgene fremlagde kort deres beretninger.

Kommentarer:

En mente at hjemmesiden burde kunne opdateres bedre og nemmere. Der ønskes uddybning af planer og visioner for det kommende år. Bestyrelsen opfordres til at udsende et visionsdokument sammen med den skriftlige beretning.

5. Fremlæggelse og godkendelse af årsregnskab 2010, se <http://dkuug.dk/genfor2011/>

Jon gennemgik det (på web) udsendte regnskab.

Der er skiftet bogholder i maj. Den nye kommer en gang per md. Vi tærer af formuen. Organisationsmedlemmer er faldende. Værdipapirer er steget 09-10 (men faldet 10-11).

Regnskabet var færdigt i oktober. Dette grundet uklarheder vedr. forpligtelser overfor EurOpen. Disse forventes endeligt afklaret i 2012. Revisorer har ingen bemærkninger hertil.

7. Forslag fra bestyrelsen:

a) Nye Vedtægter.

Henrik gennemgik de foreslåede vedtægtsændringer. Hovedideen bag ændringerne er, at det skal være muligt at forstå vedtægterne uden forudgående kendskab til foreningen.

[Herunder kort hvad der blev sagt om de enkelte rettelse. Henrik gennemgik et ark med de gamle og de foreslåede vedtægter side

om side. Et tal er paragraffen i de oprindelige vedtægter; hvis tallet har tilføjet 'ny' er det paragraffen i de nye vedtægter.]

- DKUUG er ikke og kan ikke blive en forening med begrænset ansvar. (Erhvervs og selskabsstyrelsen er forespurgt)
- Paragraffen forkortes. Flere kommenterer at det er beklageligt at 'Unix' er taget ud. En enkelt kommentar om at DKUUG ikke er en open source forening.
- Paragraffen findes ikke brugbar.
- Forenkling og præcisering af hvad et org medlem er. "4ny" Kontingent delen pillet ud og lagt i separat "5ny".
- Præcisering af hvornår et medlemsskab ophører og hvordan det håndteres. Der var diskussion om paragraffen, men det blev fremført at dette var juristernes anbefaling.
- Forenkling. Dispositionsret er flyttet.
- Tilføjelse af suppleanter. Kommentarer fra de fremmødte om at bestyrelsen 'skal' indkalde, ikke 'kan'.
- 2ny Der er en forglemmelse med antallet af medlemmer i bestyrelsen. Forslagets formulering betyder at alle er på valg hvert år.
- 10ny Er to forskellige måder at skrive det samme på. Hovedindhold er definition af hvor meget der må trækkes på formue.

Grundet tidsnød og den opdagede fejl i 8.2ny trækkes forslaget.

Det blev diskuteret hvorledes videre behandling skulle ske i praksis. Der er stemning for at der laves nye vedtægter.

8. Valg af formand. Dennis er genvalgt.

9. Valg af bestyrelsesmedlemmer

- Afstemning om tillid til Keld Simonsen.
- Valg af 4 bestyrelsesmedlemmer (de opstillede kandidater fremgår af fuldmagtsblanketten)

Pkt a:

Dennis forklarede på bestyrelsens vegne, at årsagen til pkt a er at Keld ikke er egnet til at samarbejde i en bestyrelse. (Flere på valg vil trække sig hvis Keld kommer ind.) Keld

mener at dette ikke er tilfældet.

Stemmetællere: Bryan Østergaard, Anders Kistian Andersen og Kenneth

Resultat: 14 tillid, 29 mistillid (18 tilstede, 11 fuldmagter), 3 blanke (1 tilstede, 2 fuldmagter).

Keld vil ikke respektere afstemningen, idet han ikke mener, at afstemningen er demokratisk. Keld vil gerne bruge sin medlemsret til at stille op.

Pkt b:

Kandidaterne præsenterede sig kort. Stemmetællere: Bryan Østergaard, Anders Kistian Andersen og Kenneth

Resultat: David (30) Valgt
Eskild (13)
Henrik (33) Valgt
Jakob (32) Valgt
Jon (31) Valgt
Keld (9)
Ugyldig (1)

10. Valg af revisor (bestyrelsen foreslår Svenne Krap) og revisorsuppleant (bestyrelsen foreslår Lars Josephsen)

Svenne er valgt. Lars er valgt

6. Fastsættelse af budget og kontingenter for det kommende år, herunder tilladt træk på foreningens formue.
[Dirigenten flyttede punkt 6 hertil, idet det er placeringen iflg. vedtægterne]

Jon gennemgik budget oplægget.

De der ikke har betalt kontingent per idag er smidt ud.

Det maksimalt tilladte træk er 826.000 Kr
Kontingent er uændret.

11. Evt.

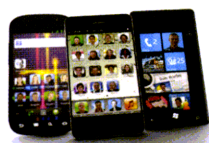
Donald talte om vigtigheden af standardisering. Herunder det politiske sigte. Han vil gerne have indlæg til bladet.










Flemming offentliggør seneste dkuugnyt på www.batmule.dk/DKUUG. Referatet lægges også her.

Dirigenten takkede for god ro og orden.

[Referat afsluttet 21:56]

Referent, Flemming Jacobsen
Dirigent, Kenneth Geissshirt



Nr.	Kursus	Pris ex. moms	Dg	Jan	Feb	Mar	Apr	Maj	Jun
Vurdering / Sammenligning af Windows Phone 7.5, Apple iOS, Android og Web Apps									
SU-899	Vurdering / Sammenligning af Udviklingsmiljøerne	4.500,-	1		10/2		13/4		15/6
SU-879	Vurdering / Sammenligning af Support og Management	4.500,-	1			16/3		11/5	
Windows 8									
MS-800	Windows 8 Install and Use (inkl. Windows 8 tablet)	 8.900,-	2		1-2/2	5-6/3	4-5/4	7-8/5	4-5/6
MS-801	Windows 8 Programmering Grundkursus	7.400,-	2		20-21/2	15-16/3	16-17/4	10-11/5	7-8/6
MS-802	Windows 8 Programmering Workshop	7.400,-	2				26-27/4		21-22/6
MS-805	Windows 8 Programmering Videregående	7.400,-	2			22-23/3			18-19/6
MS-820	Windows 8 Support	7.400,-	2		13-14/2	12-13/3	10-11/4	14-15/5	11-12/6
MS-821	Windows 8 Management	7.400,-	2				12-13/4		14-15/6
Windows Phone 7									
MS-901	Windows Phone 7.5 Program. Grundkursus (inkl. Windows Phone 7.5)	 8.900,-	2	30-31/1	27-28/2	26-27/3	30-1/5	21-22/5	25-26/6
MS-902	Windows Phone 7.5 (Mango) Programmering Workshop	7.400,-	2	2-3/1	1-2/2	1-2/3	2-3/4	7-8/5	
MS-905	Windows Phone 7.5 (Mango) Programmering Videregående	7.400,-	2			8-9/3		29-30/5	
MS-907	Win Phone 7.5 (Mango) Design and Development Application BootCamp	7.400,-	2				23-24/4		14-15/6
MS-920	Windows Phone 7.5 (Mango) Support (inkl. Windows Phone 7.5)	 8.900,-	2		6/2		10/4		4/6
MS-921	Windows Phone 7.5 (Mango) Management	7.400,-	2			19-20/3		2-3/5	
Apple IOS									
AP-901	iPhone/iPad Programmering Grundkursus (inkl. iPad)	 8.900,-	2	3-4/1 16-17/1	13-14/2 28-29/2	12-13/3	12-13/4	14-15/5	11-12/6
AP-902	iPhone/iPad Programmering Workshop	7.900,-	2	12-13/1	27-28/2		23-24/4		25-26/6
AP-903	iPhone/iPad Programmering Workshop (inkl. Mac Mini)	 11.400,-	2	12-13/1	27-28/2		23-24/4		25-26/6
AP-905	iPhone/iPad Programmering Videregående (inkl. iPhone 4S)	 11.500,-	2			26-27/3		7-8/5	
AP-920	iPhone/iPad iOS Support (inkl. iPad)	 8.900,-	2	26/1		22/3		24/5	
AP-921	iPhone/iPad iOS Management (inkl. Mac Mini)	 11.400,-	2		16-17/2		2-3/4	7-8/5	
Android									
LX-901	Android Programm. Grundkursus (inkl. Android Tablet)	 8.900,-	2	12-13/1	9-10/2 23-24/2		12-13/4	10-11/5	14-15/6
LX-902	Android Programmering Workshop	7.400,-	2		13-14/2 27-28/2		16-17/4		18-19/6
LX-905	Android Programmering Videregående	7.400,-	2			29-30/3		7-8/5	
LX-920	Android Support (inkl. Android Tablet)	 8.900,-	2		27/2		26/4		28/6
LX-921	Android Management	7.400,-	2			5-6/3		9-10/5	
Web Apps									
SU-901	Web App Programmering Grundkursus (inkl. Mobile device)	 8.900,-	2	11-12/1	13-14/2	12-13/3	10-11/4	7-8/5	11-12/6
SU-902	Web App Programmering Workshop	7.400,-	2		15-16/2		12-13/4		13-14/6
SU-905	Web App Programmering Videregående	7.400,-	2			29-30/3		31/5-1/6	
SU-920	Web App Support (inkl. Mobile device)	 5.900,-	1		20/2		16/4		18/6
SU-921	Web App Management	4.100,-	1		21/2		17/4		19/6

