

DKuug nr. 180

Februar 2016

NYT



Open Source Days
27/02 - 2016

Metropol - professionshøjskole

Sigurdsgade 26

2200 København N



Dansk Forum for Åbne Systemer

DKUUG - Unix Brugere (Linux/BSD) System administratorer

**Community for IT-specialister
og IT-interesserede.**

 **DKUUG**

Redaktørens klumme

Open source i 2016

Open Source Days (OSD) er en af de vigtigste events for computeinteresserede. Måske den vigtigste på disse kanter, for her kan man høre om nyt indenfor IT-teknik generelt. Al teknik indenfor computersoftware - og til dels hardware - er præget af nødvendigheden af at kunne dele viden hurtigt og effektivt, og det gøres bedst ved at dele kildetekster til programmer (source). Det koster ikke ret meget i tid og penge. Skær ned på administrationen og få tingene til at fungere.

Deling af kildetekster og deling af erfaringer med hardware muliggjorde hurtige fremskridt gennem sidste halvdel af 1900-tallet. Åbenheden smittede af på andre fagområder. I dag er **Open** en slags buzzword - OpenBSD, OpenSSH ja, men også Open Document, Open Learn, og Open Religion hvis man skulle få brug for det.

Open Source har vist, at "åbne" former for projektledelse kan eksistere side om side med konventionel, central ledelse. Der er mange forskellige typer ledelse i Open Source projekter, og de økonomisk vigtige har i dag sponsorer. Men udviklingsomkostningerne er ikke det centrale, det er resultaterne, stabiliteten, måden hvorpå der luges fejl. Her i bladet havde vi sidste år en artikel om Linux Foundation og hvordan der kommer flere og flere tilføjelser og rettelser mv. Artiklen brugte den årlige rapport fra Jon Corbett; han luftede tanker om at det måske ikke på længere sigt er hensigtsmæssigt, at Linux-kernen fungerer som én samlet enhed, måske skal vi til at se at der er flere forskellige "modeller" (til forskel fra fortløbende versioner), branches, eller udsplitninger, for særlige formål.

Især er der en problematik omkring størrelsen: Man kan godt lave en minimal kerne i dag, men det er ikke så enkelt som tidligere.

For nogle er det indlysende, at åbne, patentfrie standarder er den mest økonomiske model for produktion. Flere kan deltage - der skal ikke bruges tid på administration og forhandling af licenser. Programmering op imod en protokol sikrer at programmer kan udskiftes, det ene kan lige så godt bruges som det andet. Det svarer til at togvogne har samme sporvidde. (En sidebemærkning: Det ville være en lettelse, hvis Danmark og Sverige havde samme elektriske systemer til togene - Øresundstog skal skifte strømforsyning, når man er midt på Øresundsbroen.)

Indenfor IT er det indlysende: Protokoller for kommunikation skal være fælles for at give mening. Men grænserne flytter sig ustandselig, iPhone og iPod skabte et Internet for Applebrugere, et parallelsamfund, en ghetto, som alle gerne vil lukkes ind i.

Men så kom Google med et alternativt mobil-OS eller rettere en grafisk brugerflade baseret på en Linux-kerne, og alle andre producenter end lige Apple brugte Google's Android. Der er flest Android telefoner i verdenen, og de fleste applikationer - apps - kommer til begge systemer. Tænk hvis en enkelt version af en app kunne køre på alle mobiltyper!

For en, der er ung i dag, er det ret utroligt, at der blev produceret computere i 1980'erne, som ikke kunne kommunikere med andre systemer over netværk, så vi er trods alt kommet et godt stykke i retning af systemer, som kan "tale" med hinanden, selv om der er nogle ting, producenter som Apple holder for sig selv.

OSD 2016

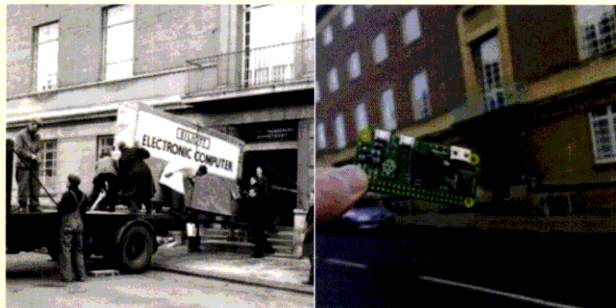
Her i begyndelsen af Februar er der begyndt at tegne sig et billede af de teknisk videregående foredrag - men OSD-2016 skal også have en række foredrag -for begyndere.

Man kan følge planlægningen på www.opensourcedays.org.

Jesper Dagaard Brouer, som er kerneudvikler for Red Hat og arbejder fra sit hjem, kommer med et emne, der er helt på forkant af den tekniske udvikling: hvordan skal kernen håndtere en forbindelse til et 100 GBit netværk. Se mere om det emne i omtalen af OSD længere inde i bladet.

Der er mange andre spændende foredragsemner.

(Fortsættes side 18)



Richard Hancock
@CanaryWorF

Follow

58 years on, Norwich City Hall finally gets a new computer:

#pizero

10:46 AM - 26 Nov 2015

3,006 retweets 2,247 likes

Dette nummer ...

... serveres der hindbærtærter og kaffe, Raspberry Pi og Java - nå ja, kaffen må I nok selv lave.

Det er selvfølgelig Opensourcedays (OSD-2016), der denne gang tager opmærksomheden. Opensourcedays ændrer sig med årene som alt andet - det er et initiativ, som strækker sig tilbage til 1999, og har tidligere lydt navnet Linuxforum.

Godt nok vidste vi her på redaktionen, at computerscience er et stort område, men når man skimmer igennem emnerne på OSD-2016 så forundres man alligevel over, hvor stort det er. OSD er selvfølgelig ikke lige så stort som FOSDEM i Bruxelles, og vor udsendte medarbejder har inspireret undertegnede til at jeg - hvis der bliver en næste gang i 2017 - absolut må med - koste hvad det vil af dyre hoteller og skrækkelige flyveture.

Men hvorfor tage til OSD når man alligevel kan læse det meste på nettet, jo, det er selvfølgelig for at se girafferne i levende live. Det er faktisk ikke så få koryfæer, der tropper op ved sådan en

lejlighed, hvilket man kan se, hvis man kan finde billederne fra de tidligere OSD og Linuxforum på website, gallery.opensourcedays.org.

Dette nummer er - som de fleste foregående - skrevet med en læserskare af begyndere og viderekomme. Det er altid svært at skrive, så det ikke er for indforstået. Skal man forklare et udtryk eller ej? Open Source, skal det oversættes til Åben Kildetekst? Skal en key oversættes til en nøgle? Skal build oversættes til bygge? Vi må vælge en passende blanding og det er ikke nogen hemmelighed at den blanding ikke ligner almindeligt dansk, men det er vilkårene for et fagsprog.

En ofte fremført anke mod Unix var, at de almindelige utilities (godt ord igen) havde kryptiske navne, men det skyldtes jo simpelthen at man ville undgå overload af generiske termer, - så print på skærmen blev til echo (det er en af de gode) og listfiler blev til ls - som udtales ell-ess. Om det er en gevinst for IT-udbredelsen at ordene ofte er låneord fra engelsk kan man somme tider komme i tvivl om, men så må man se det positive - man får lært lidt engelsk samtidig.

Donald Axel

DKuug-NYT er medlemsblad for DKuug, foreningen for Åbne Systemer og Internet
Nr. 180 - Februar 2016

Udgiver:

DKUUG
Fruebjergvej 3
2100 København Ø
Tlf. 39 17 99 44
email: blad@dkuug.dk

Redaktion:

Donald Axel (ansvarshavende)

Forsidecredits:

Redaktionen og Google maps

Design og layout:

DKUUG/Donald Axel med LibreOffice

Annoncer:

pr@dkuug.dk

Tryk:

Lasertryk i Aarhus

Oplag:

400 eksemplarer

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

Eftertryk i uddrag med kildeangivelse er tilladt.

Deadline for nr. 181: Fredag d. 15. April 2016.

Medlem af Dansk Fagpresse

DKUUG-Nyt

ISSN-1395-1440



Vores møder og foredrag holdes - med mindre andet udtrykkeligt angives - på vores adresse:

DKUUG
SYMBION
Fruebjergvej 3
2100 København Ø

Hvis man kommer lidt før, er der tid til en snak på kontoret. DKUUG bor i en virksomhedsfarm, Symbion, hvor der er åbne døre indtil kl.18 eller 19 (afhængig af mødetidspunkt). Efter den tid har vi på foredragsaftener en vagt ved døren.

INDHOLD:

OSD 2016: fra 100 Gbit til Fuzzing

af Donald Axel 4

Debian grundlægger Ian Murdock mindeord .. 6

E dokumenter i retssalen - en røverhistorie

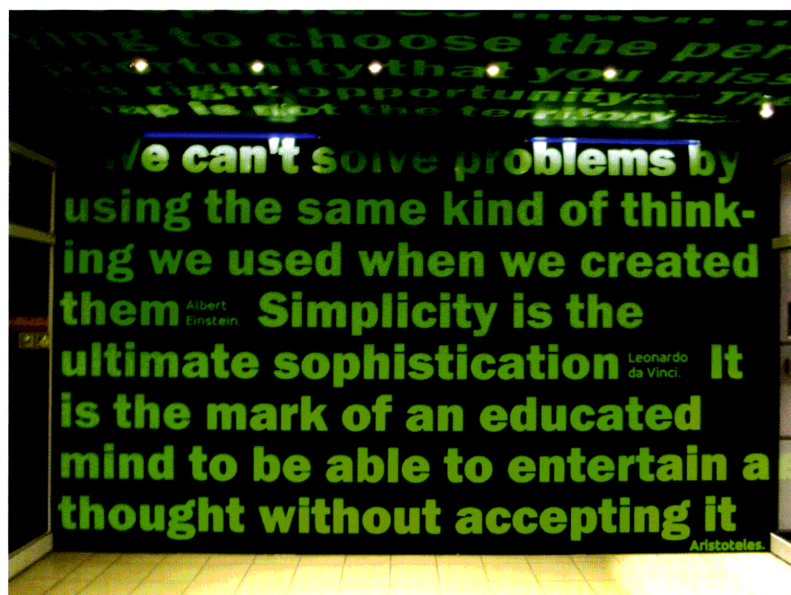
af Donald Axel 8

Raspberry Pi & Java

af David Askirk 10

FOSDEM 2016 - Januar Bruxelles 16

Kommando(linie)centralen 19



Arrangementer:

Opensourcedays 2016: Lørdag den 28. Februar: kl. 10 til 18 -

Billet 150,- købes på opensource.org

Onsdage : møde på kontoret fra kl.18 ca. - somme tider 17 eller før.

Vi modtager gerne indlæg til bladet.

DKUUG udgiver foreningsbladet DKUUG NYT 3 eller 4 gange om året - og oftere, hvis der er stof nok. Bladet fremstilles indtil videre med LibreOffice. Vi kan godt hjælpe med brug af Libre Office og dele templates mv.

Libre Office er nævnt inde i bladet i forbindelse med FOSDEM i Bruxelles d.30-31.Januar, hvor Jos van den Oever fortalte om brugen af ODF i Europa.

"The Internet? Is that thing still around?"

(Homer Simpson)

OSD-2016: fra 100 Gbit til Fuzzing

Af redaktionen



Open Source Days
27/02 - 2016

Der bydes på teknisk avancerede indlæg for eksperter og introduktioner for begyndere -

Opensourcedays 2016 afholdes den lørdag den 27. februar i professionshøjskolen **Metropol** - alias Metropolitan University College - i Sigurdsgade 26, 2200 København N. Sigurdsgade løber fra Lersø Parkallé i svinget nærvæd Jagtvejen sådan cirka til Tagensvej - i fremtiden vil der være en metro-station vd Tagensvej og man kan på forsiden se gadebilledet hvor kranen i baggrunden har med bygning af metro'en at gøre.

Billet koster 150 kr. og kan købes på websitet **opensourcedays.org**.

OSD arrangerer et spor for begyndere, fordi det bliver mere og mere tydeligt at der er behov for introduktion til Open Source og mere generelt, introduktion til baggrunden for Open Source: Større effektivitet ved bedre udnyttelse af programmørers viden, og ikke mindst vidensdeling; dertil kommer alternativ projektledelse, og det morsomme ved ikke være afhængig af hvad andre synes man skal gøre.

Der er mange slags mennesker, som gerne vil vide noget om de grundlæggende ting indenfor Open Source og datalogi, computer science, og derfor kan man håbe på at der vil opstå frugtbare diskussioner og inspirerende sammentræf på OSD-2016.

Metropolitan University College

Stedet er nyt i forhold til de foregående år. OSD har fået en aftale med professionshøjskolen **Metropol**, **www.phmetropol.dk/** om lokaler og faciliteter, ikke mindst kantinen, som vil være åben under hele forløbet. Efter foredragene, som begynder kl.10 og slutter henad kl.18, vil kantinen stadig være åben og derfor er der mulighed for social omgang med både nye og gamle venner og bekendte.

I skrivende stund spænder OSD vidt i emner, men tidsplan og alle tracks er ikke på plads endnu, så der vil komme flere. DNYt må henvise til websitet **opensourcedays.org** for de foredrag, som kommer til i løbet af begyndelsen af Februar.

Kerneudvikler fortæller om sit arbejdsområde

Det er ikke hver dag, at man får adgang til at høre en af de programmører, som er med til at lave Linux-kernen (eller andre operativ-system kerner). Vi er måske lidt forvante i Danmark, for vi har nogle stykker af slagsen. Jesper Dangaard Brouer er kerneudvikler - og er hoved-kerneprogrammør - *principal kernel engineer* - hos

Red Hat, og er med i det team, som vedligeholder Linuxkernens netværks-stak. Han arbejder i sit hjem udenfor København. Han har deltaget i mange konferencer eksklusivt for udviklere, blandt andet Netconf, Netfilter Developer Workshops. Han er også del af det hold, som vedligeholder netfilter.org - de funktioner, som bruges til bl.a. firewalling.

Efterhånden som netværkshastighederne kommer op over 10 Gbit/s - til 40 Gbit/s og snart til 100 Gbit/s - så vil hastigheden, hvormed pakker ankommer forøges, og som en følge af det er der kun 6.7 nano-sekunder (1000dele millitedele) pr pakke. Foredraget forklarer problemerne som kernens netværks-stak dermed bliver udsat for og beskriver nogle af de strategier, man har tænkt på at bruge til at løse den opgave. Hvis det skal kunne lade sig gøre, må man frakoble de mekanismer i CPU og memory, som tager ekstra tid, d.v.s. memory management.

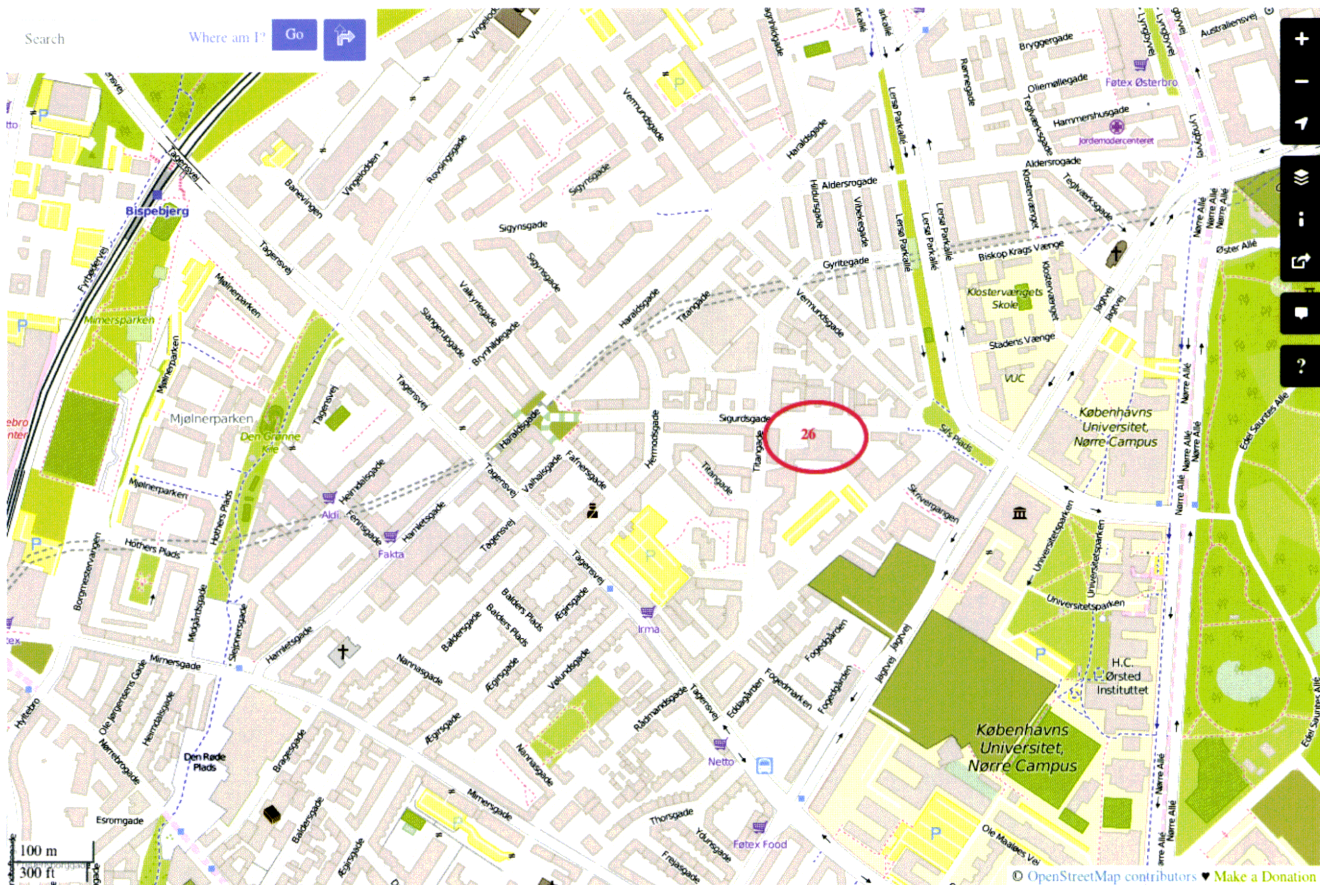
Slår man op i Wikipedia på TCP Offload Engine, TOE, finder man ud af at uddelegering af fx. checksum for TCP/IP pakker har været et problem i Linux-kernen - tidligere, ikke mere. Det ligger i netværksdriveren nu, d.v.s. at CPU'en ikks skal gøre ret meget for selve TCP/IP beregning af checksum (og længde mv). Nu er CPU'et blevet hurtigere; alligevel ville det jo ikke være så rart hvis CPU'en bliver overbelastet fordi man henter data med høj hastighed. Linux netværksdrivere benytter TOE (og har gjort det siden 2002).

Men med springet fra 10 til 40 til 100 Gbit/s er der alligvel tale om nogle ret så alvorlige udfordringer! Tænk på det på denne måde: En CPU som kører 2 GHz laver ikke en instruktion pr. Hertz, den laver ikke 2 tusinde millioner (mia) instruktioner i sekundet - i hvert fald ikke hvis det er en Intel, for den bruger mellem 2 og 10 cycles til at udføre en instruktion - og nogle få tager mere end 20 clockcycles (division).

Men selv hvis CPU'en kan lave 2 mia instruktioner i sekundet er der et stykke vej til at håndtere 100 mia bits. Også hvis vi håndterer dem i 64-bit ad gangen. Måske kan det gå, hvis vi kan smide pakkerne ud til hver sin core. Måske kræver det ekstra pilleri ved memory management - mon ikke det er sådan, det hænger sammen! Så ja, det bliver spændende at høre, hvad Jesper Dangaard Brouer har at fortælle om det.

For den praktisk anlagte system administrator

Monitorering af systemer, besked om fejl, eller temperaturforandringer, disk-fyldningsgrad, memory-swap, alle disse ting er afgørende for om et belastet IT-system yder optimalt.



Måling af performance har altid været et vigtigt emne, i dag fordi det er forudsætning for stabil drift - tidligere var det også en bedømmelse af omkostningerne, fordi man skulle betale pr. CPU operation ud over leje eller køb af mainframe systemer (men prisaftaler vedrørende de store IT-anlæg har altid været hemmelige).

Et af mine favorit emner er musik og hardware og software. Lyde og udtryk kan flytte opmærksomheden over i en anden verden, så at sige.

Niklas Nisbeth holder foredrag om Axoloti - et kort, som kan generere patches/envelopes til lyde. Axoloti har stereo in/out, audio analog til digital-til-analog, og en microcontroller som er god til digital audio-signal processing. Forbindes til en PC eller Mac og så kan man begynde lyd"patching" med Axoloti patcher, læne sig tilbage og nyde resultatet.

Rent teknisk laver man synthesizer lyde ved at opstille en "kæde" - med Axoloti software kan man se hvad man laver (man får med andre ord noget på skærmen).

Både hardware, software og firmware er Open Source - og det er nyt. Jo, der er en synthesizer Fluidynth, som er integreret i nodeskrivningsprogrammet *Muscore*, men man kan ikke pille ved selve lydene gennem nodeskrivningsprogrammet (som iøvrigt fortjener mere ros og mere omtale).

Alle kender Google. For 15 år siden var det nyt og næsten ukendt. Søgning var koncentreret om websites, siden kom e-bøger, scanning af bøger til Google-ebooks, søgninger, som også omfattede bøger på Amazon og Googlebooks, og søgninger af flere og flere slags.

Peter Makhholm fortæller om ElasticSearch, som blandt andet tilbyder fuldtekstsøgning i JSON dokumenter. Dette gør ElasticSearch velegnet til søgning i tekst, hvor traditionelle databaser som MySQL og PostgreSQL kommer til kort. Med udgangspunkt i en tænkt webshop introducerer Makhholm ElasticSearch og gennemgår nogle af mulighederne.

The Fuzzing Project

Hvis et program kan forbedres og er stabilt så brugerne ikke oplever udfald eller "frysning" (crash, blå skærm osv) så har programmet meget bedre mulighed for større udbredelse og dermed tilslutning fra både brugere og programmører. Der er mange måder, man kan forbedre kodekvaliteten, og i et foredrag med titlen *The Fuzzy Project (Det Uldne Projekt)* giver Hanno Böck et indblik i hvordan man kan generere alle mulige varianter af input for at teste et programs stabilitet.

Idéen er at man giver *fuzzing* programmet en kommandolinie, med et program, som skal testes, med mindst en inputfil. Fuzzeren prøver at korte denne inputfil ned og begynder derefter at ændre på input filen. Hvis test-programmet crasher, har vi fundet en ny bug.

Det er også en måde at finde sikkerhedsfejl i test-programmet. En American Fuzzy Lop er en art kanin, og dette fuzzing engine program kan generere "låddent input".

Debian grundlægger Ian Murdock mindeord

28. april 1973 -- 28. december 2015

Navnet er en sammenkædning af Ians navn og hans kæreste og senere kones navn, Debra Lynn. Ian arbejdede med Linux fra 1993, og dannede navnet Debian for den Linux-opsætning, som han havde lavet. Ian og Debra blev enige om at andre også skulle have glæde af opsætningen, det var noget, som måtte deles. Forud var der Slackware som Murdock imidlertid syntes havde for mange fejl. Slackware var disketter med både Linuxkerne og de vigtigste programmer, fil-utilities, ed, sed, awk, og den slags. Også GNU C kompilatoren måtte man have på en anden diskette. Netværk havde man ikke.

Det tidlige Linux kunne installeres på en harddisk og kernen kunne allerede dengang håndtere store harddiske, idet Linus Torvalds lod kernen installere sin egen BIOS (1GB var en pænt stor harddisk i 1993, men ikke alle BIOS'er ville kendes ved diske, der var så store.)

1994 skrev Ian Murdock et Debian Manifest - og så var distributionens karakteristik fastslået: Det var et fælles projekt, alle kunne bidrage. Det var stadig noget som kun var for særligt teknisk interesserede, for der var jo ikke nogen grafisk front.

Det havde til gengæld Windows-3.1 - men det var ikke ægte multitasking.

Det specielle ved Linux systemet, det, som tiltrak fagfolk som bl.a. Eric Raymond, var at operativsystemet viste at man kunne bruge en Intel 386 box med memory management med samme tekniske fordele som de store hidtil ubetalelige systemer (Altos og lignende) som var på markedet fra midtferiserne. Så kom denne overraskelse: Linus Torvalds lavede fra sit studerkammer en 32-bit kerne med pre-emptive multitasking og memory protection, d.v.s. at et program får tildelt nogle millisekunder og derefter bliver afbrudt, - CPU'ens tilstand bliver gemt, forskellige counters opdateret osv og der bliver kørt en algoritme, som vurderer hvilke andre programmer, der har mest behov for CPU-tid. (Meget forsimplet beskrivelse).

Der var godt nok andre systemer, som kunne køre flere programmer samtidig, vi er nogle stykker, som i 80'erne kørte Xenix, et Unix operativ-system, som kunne lave preemptive multitasking. Også CCPM, senere Concurrent Dos viste at PC-hardware kunne multitask, men programmerne havde adgang til hele memory - et uvorent program kunne crashe hele systemet.

En IBM PC forhandler sagde på det tidspunkt, at en PC-Dos da sagtens kunne multitask - den kørte mange programmer efter hinanden, sekventiel multitasking! Men Linux (og Debian) var jo som sagt et rigtigt multitasking system.

Debian distributionen i sin grundform (den "ægte" Debian) bliver af mange set som operativsystemet, der er **rigtig** Open Source, fordi der skelnes mellem applikationer, som er **free** og **non-free**. Der er flere udviklere, som er imod at Debian arkiverne omfatter non-free programpakker.

Begejstringen for Debian er stor fordi Debian er stabilt som en klippe. Det var jo også Murdock's oprindelige mål med projektet. (Man skal dog lige passe på at vælge de almindeligste pakker og de almindeligst udbredte ting - der kan være sjældent brugte programmer, som egentlig ikke burde være med i arkiverne). Man skelner mellem pakker, som er stabile (*stable*) og under afprøvning (*testing*) og helt ukendte med hensyn til stabilitet (*unstable*) - for den sidste pakketype tager man selv ansvar og hvis man vil, er man velkommen til at rapportere fejl og mangler.

Uden Ian Murdoch var jeg formodentlig endt som Windows administrator, tweetede en Debian-tilhænger @sadservers.



Foto af Ilya Schurov, Computerra Weekly

Afledte distributioner

Der er mange distro'er, som går ud fra Debian og tilføjer en overbygning med de elementer, som de nu synes er vigtige.

Distro-watch har i skrivende stund 126 aktive Debian derivater.

Den kendteste er nok Ubuntu, som er finansieret af Mark Shuttleworth, og som lægger mere vægt på brugerens valg og bl.a. tilføjer muligheden for at vælge mellem flere Grafiske User Interfaces (GUI). Debian kører på "alt", men det gør Ubuntu også: *Ubuntu is an open source software platform that runs from the cloud, to the smartphone, to all your things.*

Ledelse

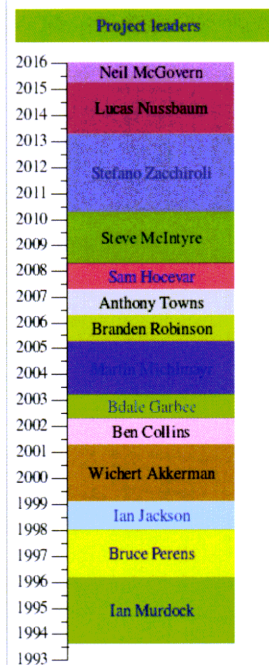
Allerede i 1996 overlod Ian Murdock ledelsen af Debian til en anden, Bruce Perens, som er den, der sammensatte betegnelsen "Open Source" som en slags alternativ markedsføring, idet mange af de store IT-brugere var bekymrede over udtrykket "Free Software", ud fra en formodning om at når noget var gratis (free) så var det ikke værd at eje. Richard Stallman forklarer at "free er som fri i frihed", free as in freedom, for at undgå den misforståelse, men som bekendt er Open Source slået an som betegnelse for vidensdeling - det er mere præcist fag-orienteret.

Men Bruce Perens blev også afløst og i Wikipedia kan man se listen over project ledere i Debian - der er til dato 14.

Murdock's idéer

Pakkesystemet i Debian byggede på de almindelige Unix standard-tools (standarden blev beskrevet og fastholdt i en ISO standardudgivelse **Posix**) - ganske vist i forbedrede udgaver efterhånden som GNU udviklingsværktøjer og mere memory og gode udviklere fik flere gode idéer. Grundlæggende er en Debian pakke et arkiv dannet ud fra **ar(1)** specifikationen og indeholder **tar** filer, som så igen indeholder henholdsvis instruktioner og programfiler og det er instruktionsfilerne, som har selve installations-kommandoerne - og så udfører debian-pakkeprogrammet instrukserne.

Murdock sagde om debian-pakkesystemet *at det ikke er lavet for at administrere software, men for at facilitere samarbejde.*



The Project Leader is the public face of Debian and defines its direction. ^{[140][141]}



Download Debian 8.3
(32/64-bit PC Network installer)

Debian is a [free](#) operating system (OS) for your computer. An operating system is the set of basic programs and utilities that make your computer run.

Debian provides more than a pure OS: it comes with over 43000 [packages](#), precompiled software bundled up in a nice format for easy installation on your machine. [Read more...](#)

About

- » Social Contract
- » Code of Conduct
- » Free Software
- » Partners
- » Donations
- » Contact Us

Help Debian

Getting Debian

- » Network install
- » CD/USB ISO images
- » CD vendors
- » Pre-installed

Pure Blends

- Debian Packages
- Developers' Corner

News

- » Project News
- » Events

Documentation

- » Release Info
- » Installation manual
- » Debian Books
- » Debian Wiki

Support

- » Debian International
- » Security Information
- » Bug reports
- » Mailing Lists
- » Mailing List Archives
- » Ports/Architectures

Miscellaneous

- Site map
- Search
- The Debian Blog
-
-

The [latest stable release of Debian](#) is 8.3. The last update to this release was made on January 23rd, 2016. Read more about [available versions of Debian](#).

Getting Started

- If you'd like to start using Debian, you can easily [obtain a copy](#), and then follow the [installation instructions](#) to install it.
- If you're upgrading to the latest stable release from a previous version, please read [the release notes](#) before proceeding.
- To get help in using or setting up Debian, see our [documentation](#) and [support](#) pages.
- Users that speak languages other than English should check the [international](#) section.
- People who use systems other than Intel x86 should check the [ports](#) section.

News

RSS

[23 Jan 2016] [Updated Debian 8: 8.3 released](#)

Debians website, forsiden

Hvor mange deltager i udvikling af Debian?

Historical population		
Year	DD	±%
1999	347	—
2000	347	+0.0%
2001	?	—
2002	939	—
2003	831	-11.5%
2004	911	+9.6%
2005	965	+5.9%
2006	972	+0.7%
2007	1.036	+6.6%
2008	1.075	+3.8%
2009	1.013	-5.8%
2010	886	-12.5%
2011	911	+2.8%
2012	948	+4.1%
2013	988	+4.2%
2014	1.003	+1.5%

Source: [Debian Voting Information](#)

Der er en stadig strøm af ansøgere til at blive *udvikler* i Debian-projektet.

De må først igennem et baggrundscheck, som fastslår identitet, motivation, forståelse af principperne og teknisk kompetence. Den proces er blevet mere og mere vanskelig at komme igennem.

Man kan selvfølgelig også trække sig tilbage og blive *emeritus*.

Hvert softwareprojekt (programpakke som fx. coreutils) har en maintainer (en, som har ansvaret for at der laves Debian pakker af den software som kommer fra programmørerne.)

Ledervalg og diskussioner

Der holdes en årlig konference, DebConf, med foredrag og workshops for udviklere og tekniske brugere. Fra 1999 blev projektlederen valgt for et år.

Ud over alle grænser

Den 2. december 2015 annoncerede Microsoft at de vil tilbyde Debian GNU/Linux som den anbefalede distribution på on Azure cloud platformen.

Ian Murdock's farvel

I 2006 blev Murdock udnævnt til teknisk chef for *Free Standards Group*, og chef for Linux Standard Base Workgroup. Han fortsatte som teknisk chef for Linux Foundation der blev dannet ved sammenslutning af Free Standards Group og Open Source Development Labs.

Allerede forinden havde han sagt *“Min endelige test for hvorvidt Debian er lykkedes er: Kunne grundlæggeren træde væk og ville projektet gå videre, for det er det eneste tidspunkt hvor man ved at projektet har fået sit eget liv.”*

E-dokumenter i retssalen

Et af de morsomste eksempler på forfalsket e-mail - og en introduktion til DMARC

af Donald Axel



En introduktion til mailsikring

De fleste af os - både her og der - har opdaget at gmail er sikrere, bedre og billigere. De er gode til at filtrere svindlere fra.

Gmail har 900 millioner brugere, sådan ca. - antallet offentliggøres ikke præcist :)

De fleste af os bemærker ikke længere e-mail fra Nigeria med anmodning om at dele en kæmpegevinst hvis blot vi betaler et par tusinde dollars til den og den konto. Desværre er den slags mail-svindler eksempel på at den utroligt lette adgang til at sende mail har lettet adgangen til naive ofre - det koster næsten ikke noget at have netforbindelse og sende tusindvis af mail, problemet er at få valide adresser. Hvis man sender en million mail og een af dem giver gevinst på 5-6000 kr. så har man allerede overskud.

Den slags mail-scam forudsætter, at man kan sende mail med et navn og en stillingsbetegnelse, som lyder tillidsvækkende. Det er faktisk ikke sværere end at skrive afsender bag på en papirkonvolut.

Det er en kendt ting, at afsender på en e-mail kan ændres, enten ved at bruge en option til de fleste mail-programmer, - eller ved at patche de såkaldte envelope filer.

```
sendmail -f rich@plan.com you@gmail.com
```

Ovenstående eksempel vil dog blive bremsat af gmails authentication - der forlanges password hvis man vil sende til gmail serveren (ovenstående eksempel er til lokal-mailservere og er selvfølgelig kun demonstration); og når man har afleveret noget til gmail kommer login-kontoen på som afsender.

Til en demonstration erstatter jeg modtageren med en konto lokalt:

```
sat2:/ #sendmail -f rich@plan.com fox-40@saturn.saxen  
Hello this is only local mail.
```

```
sat2:/ #
```

Så ser vi efter om der er kommet mail:

```
sat2:/ #Mail -f /var/mail/fox-40  
Mail version 8.1.2 01/15/2001. Type ? for help.  
"/var/mail/fox-40": 3 messages 1 new  
 1 root@saturn.saxen Fri Feb 05 19:56 14/456  
 2 donaldus-arsenal@ Fri Feb 05 20:09 15/518  
>N 3 rich@plan.com Fri Feb 05 23:49 14/460  
&
```

Nr. 3 mail er ny ankommet og minsandten!

Den er fra Rich (Obs! det ses på promptet på kommandolinien at det er root, administrator, som gør det. De fleste mail-servere og mailklienter blokerer for den slags; meningen med det er naturligvis at administrator kan expedere andres mail videre.)

```
From rich@plan.com Fri Feb 05 23:49:29 2016  
Return-path: <rich@plan.com>  
Envelope-to: fox-40@saturn.saxen  
Delivery-date: Fri, 05 Feb 2016 23:49:29 +0100  
Received: from root by saturn.saxen with local (Exim 4.80)  
(envelope-from <rich@plan.com>)  
id 1aRpBo-000416-Ha  
for fox-40@saturn.saxen; Fri, 05 Feb 2016 23:49:29 +0100  
Message-Id: <E1aRpBo-000416-Ha@saturn.saxen>  
From: rich@plan.com  
Date: Fri, 05 Feb 2016 23:49:29 +0100
```

Hello this is only local mail.

Vi kigger på mail med **Mail**-programmet (man kan bruge hvadsomhelst, det vigtigste er at vi ser hele headeren) - og det ser jo unægtelig ud som om både mail-afsender og retur-adressen er rich@plan.com

Med **whois**-programmet kan vi se, hvem der ejer domænet **plan.com** - og det er ikke Donald Axel:

```
>N 3 rich@plan.com Fri Feb 05 23:49 14/460  
& v3  
& x  
sat2:/ #whois plan.com
```

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

```
Domain Name: PLAN.COM  
Registrar: MONIKER ONLINE SERVICES LLC  
Sponsoring Registrar IANA ID: 228  
Whois Server: whois.moniker.com  
Referral URL: http://www.moniker.com  
Name Server: BONNIE.NS.CLOUDFLARE.COM  
Name Server: IAN.NS.CLOUDFLARE.COM  
Status: ok https://www.icann.org/epp#OK  
Updated Date: 15-dec-2014  
Creation Date: 18-mar-1997  
Expiration Date: 19-mar-2016
```

>>> Last update of whois database: Fri, 05 Feb 2016 22:53:12 GMT <<<

Der er mere output fra whois (som er klippet væk her), det vigtigste her er at Registrar for Domain Name plan.com er Moniker Inline Services i USA. Kontakt mail-adresse (står længere nede og er ikke med på skærm-dumpet) er abuse@moniker.com

Whois kan slå op i forskellige registre. De forskellige top-domæner (fx .com og .dk) registreres forskellige steder.

Hvis det var en mail som vi havde modtaget udefra, burde vi nok advisere moniker.com om at deres domæne (navn) blev misbrugt.

Hvis processen med at finde ud af om domæne-navne er "gode" eller bliver misbrugt automatiseres med en liste - en *white-list* over domæner, som vi gerne vil modtage mail fra, så kan en masse spam undgåes. Men antallet af mail-domæner er jo enormt, og en liste på flere hundrede millioner domæne navne vil være for vanskelig at håndtere, derfor nøjes man ofte med en *black-listing*.

På den måde kan man nemt sortere det værste spam fra. - Der er mange måder at få fat i en *blacklisting*, som man kan bruge med sin mailservere - det kan man finde mere om på Wikipedia:

[en.wikipedia.org/wiki/Blacklist_\(computing\)](http://en.wikipedia.org/wiki/Blacklist_(computing))
og [Comparison_of_DNS_blacklists](http://en.wikipedia.org/wiki/Comparison_of_DNS_blacklists) med mere.

Men der er også mange flere måder at identificere spam og scam/fup og stoppe det. Her er det imidlertid blot afsenderdomænet, som interesserer, fordi det har været brugt til nogle af de mere utrolige sager, hvor mail skulle være bevis i en retssag.

Falske e-mail i en 25 millioner dollar retssag

MRI scannere (**M**agnetic **R**esonance **I**maging) er en moderne teknik som kan give lægerne et godt billede af de indre organer. Problemet er, at man skal ligge stille under scanningen. Hjertet bevæger sig jo hele tiden, så det kan man ikke scanne på den måde.

Men så er der et firma LBDS Holding Company i Texas, som efter sigende skal have udviklet en teknik, som løser problemet, og som vil bevogte denne teknik som intellektuel ejendom og have penge for den, når den bliver inkorporeret i MRI scannere.

Et koreansk firma, ISOL, bygger scannere og de importeres af LBDS og sælges i USA under LBDS' varemærke.

Men ifølge LBDS har ISOL brugt denne nye teknologi uden tilladelse til sine egne maskiner, der sælges under ISOLs eget varemærke.

Efter en 6 dages jury-retssag blev ISOL dømt til at betale en bøde på \$25 millioner for tabt fortjeneste.

Men efter nogen tid anklager ISOL så det amerikanske firma for at have svindlet i retssagen, LBDS havde fabrikeret falske mail, som var brugt i bevismaterialet, og som gav indtryk af at ISOL havde indrømmet.

Det var selvfølgelig en bombe under retssagen.

Resten af historien er ikke mindre mærkelig, - man kan selvfølgelig forstå at en advokat, der har brugt forfalskede dokumenter som bevis må rette op på det, ellers mister han sin bestalling og sit gode omdømme, men LBDS mente jo stadig at være i sin gode ret og var ikke så begejstrede for at tilstå - de havde jo grundlæggende set ret, mener de.

Men den går selvfølgelig ikke i en retssag, så den ærlige advokat måtte på egen hånd gå til domstolen, forklare sagen og bede om at blive fritaget for det videre forløb. *Sådan en klient kan en ærlig advokat ikke have!*

Metoden

Men det groteske i historien er måden hvorpå de her mail var forfalsket.

LBDS havde fået oprettet et nyt domæne, cernerinc.com og havde sendt mail til sig selv fra det domæne. Cerner er et rigtigt health-care firma (privathospitaler) men bruger domænenavnet cerner.com. Det var de forfalskede mail fra cernerinc.com, der blev brugt som dele af bevismaterialet.

Hvis man nu søgte (det er ændret nu) med whois på cernerinc.com ville man kunne se at det ikke var firmaet Cerner, som havde dette domæne.

Wire fraud

De kan være meget kreative, disse svindlere. En anden svindel som omfatter falske Cerner personer er af typen *wire fraud*, finansiel svindel om anvender elektronisk kommunikation (telegrammer, fax, e-mail mv.)

De falske Cerner har også givet sig ud for at sælge en MRI scanner for en million dollar og har fået betalinger tilsendt til deres falske Cerner bank-konto.

Svindelen er omfattende - alene at holde sammen på det må kvalificere til en højere akademisk grad:

Ifølge en vidneserklæring i forbindelse med sagsanlæg mod Albert Davis har Davis og hans sammensvorne givet sig ud for Cerner ansatte, læger, investeringsrådgivere, de har oprettet et falsk Cerner firma, en falsk Cerner bank-konto, og som sagt Internet domæne, e-mail, lejet kontorer med falsk Cerner-adresse, betalt for mobiltelefoner registreret som lokale, falske prislister, og falske fakturaer.

Men mest imponerende (!) er det at de skabet mindst 70 falske virksomheder, registreret i Texas, Colorado, Nevada, Wyoming, Washington, Delaware og Florida.

Kan mail bruges i retssager i dag?

Hvis en mail kan verificeres af flere uafhængige eksperter som afsendt af en given person og modtaget af en anden i denne sammenhæng rigtig person, så kan den indgå som bevismateriale, - men det er faktisk ikke så nemt med mindre man bruger private/public key kryptering/signering.

En mail kan forfalskes. Hvis afsender erkender at have sendt den eller der er vidner på at vedkommende er ophavsmand, og modtager erkender at have modtaget den, så kan en mail indgå i en retssag, ellers er det "von hörensagen" - rygter.

Så enkelt er det. En mundtlig aftale er også gyldig - hvis den bekræftes af vidner, så er den lige så gyldig som en "papiraftale".

Et dokument af papir kan også forfalskes, men her har man dog altid haft mekanismer til at verificere. Ved større handler er der underskrift, tid og sted, skøder kan tinglyses, aftaler kan stemples og noteres hos en notarius publikus; der kan være vidner, som skriver under, og som kan indkaldes til retssager.

Hvad er Message-ID

Den underlig streng af tegn på en mail-header tilføjes af mail-klienten for at en mail skal have en global unik identifikation, det gør det lettere for mailservere at opdage hvis der er to af samme mail.

Message ID skabes simpelthen på følgende måde:

Først et "<" (mindre end-tegn)

Få tidspunktet i højeste opløsning (sekunder er OK).

Generer 64 random bits.

Konverter de to tal (tid og random) til base 36 (0-9,A-Z) og sæt et punktum mellem de to tal.

Tilføj "@" - det gør venstre side af message-ID til ca. 21 characters, det er ikke så slemt ...



Tilføj til sidst det fulde domæne navn på den lokale host eller host-navnet i afsenderens retur-adresse.

Til sidst tilføjes ">" (større-end tegn).

En dame blev meget stødt over en fejlmeddelelse, som kom ud af Linux-systemet. Der var en block read fail på et FAT filsystem, og den prosariske besked var "FAT bread failed". Det opfattede hun som om, at system mente, at hun var fed og lavede fejl.

Raspberry Pi og Java

Af David Askirk

Introduktion til Pi

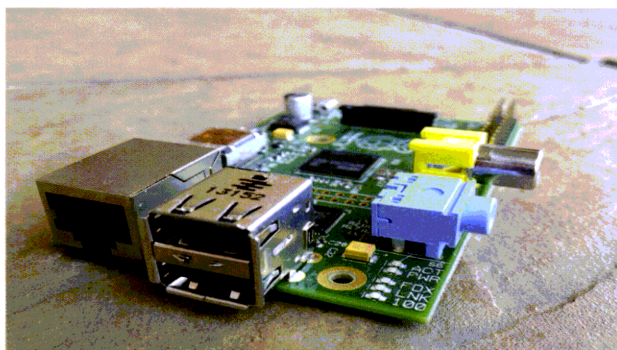
Raspberry Pi er en computer fra England. Den er rettet til undervisningsbrug og muligheden for at lære børn og unge om computer og datalogi (i gamle dage EDB).

Da den første Pi kom blev den revet væk, ikke så meget til undervisning, men til brug af "hackere"¹. Det der skete var nemlig at computeren kun kostede 35\$.

Når computeren samtidig havde netværk, IO-porte og kørte Linux gjorde at den allerede var et hit i maker og hacker¹ communityet inden den var kommet på gaden. Da den blev lanceret kunne de ikke følge med efterspørgselen og der blev solgt 1 million det første år, de havde selv regnet med 10.000 det første år.

I Pi familien er der flere forskellige slags. B, B+, A, A+, Pi 2 B og nu også Zero. I denne artikel er det Pi 2 B der bliver brugt, men koden og koncepterne er det samme på alle Pi.

¹Hackere: her menes folk der kan finde nye løsninger der bruger eksisterende teknologier på nye måder



Raspberry Pi er hardware specifikation, oprindeligt en model A og en model B (billedet), som havde flere IO interfaces, nu også en Pi2 Model B, som er bagudkompatibel med model B (Credit: Justin417 - Own work, CC BY-SA 3.0, \$3)

De original Raspberry Pi and Raspberry Pi 2 fremstilles gennem udlicitering af Newark element14 (Premier Farnell), RS Components og Egoman. Hardware er den samme hos alle producenterne.

Varianterne er mange. Den ældste række hedder i dag Pi 1. og af dem var der A og B, så kom der en A+ og B+ med flere IO porte. Pi 2 er med en nyere Quad core ARM og oprindeligt kun som model B med mange tilslutninger.

Alle Raspberry Pi er med samme video-core. De har en sikker SDHC eller MicroSDHC (A+ og B+ og Pi 2) for SD kort. Man kan boote fra SD kort - og det bruges som hukommelse.

Hov - så kom der også lige en Pi Zero!

Raspberry Pi Zero - Max 1 Pi Zero Per Order!

Pi Zero Complete Starter Kit	Out of stock	Notify me
Pi Zero + Adaptors + PiBow Zero Case	Out of stock	Notify me
Pi Zero + Adaptors	Out of stock	Notify me
Pi Zero only	Out of stock	Notify me

THIS PRODUCT IS LIMITED TO 1 UNIT PER ORDER.



Installer og opsætning af en pi

Hvordan kommer man så gang med en Pi computer? Det er nemt. Den er blevet designet så den kan køre med de ting man har derhjemme i forvejen. En Pi snakker HDMI, bruger et SD kort og kan bruge et usb keyboard og usb mus. Som strømforsyning bruges en mobil-oplader.

Pi'en er blevet designet som en efterfølger til BBC Micro og lignende maskiner fra 80'erne som også blev tilsluttet TV'et. Dvs, den bruger den ting, man har derhjemme.

Derfor kommer en Pi heller ikke med en kasse som udgangspunkt. Dette giver mulighed for at man kan få lov til at se hvordan sådan en dims ser ud, uden at det bliver magisk, eller på anden måde noget som er skjult for det almindelige øje. Det kan hjælpe på forståelsen af en computer at man kan se hvordan stik, kredse og andre komponenter ser ud.

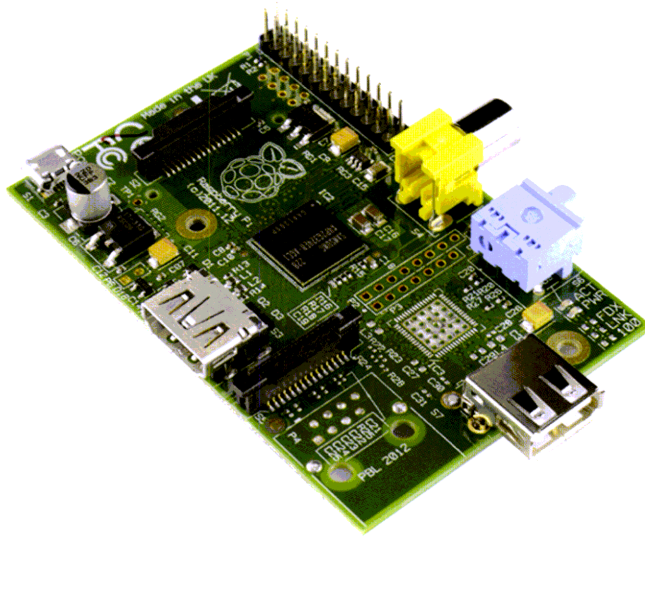
Muligheder med en pi

En Pi er en fuld computer der har en ARM processor. Udover ram, CPU, skærm og andre ting, har Pi'en en GPIO header (General Purpose IO). Det er denne header som gjorde den så populær da den først kom. Da den kom kunne man pludselig få en computer, som der var et stort community bag, billig, kørte linux og kunne programmeres med python.

Når der er disse ting til stede er der ingen grænser for hvad sådan en Pi kan bruges til. Folk bruger dem som en billig hjemmeserver, NAS-bokse, hjemme overvågning og styring af hjemmet, garagedøre, men også i mere aparte projekter som telefonbokse, robotter og lignende.



Raspi-mower Dalek - fra Dr. Who-TVserie?



Raspberry Model A (Credit: SparkFun Electronics from Boulder, USA - Raspberry Pi - Model A, CC BY 2.0, \$3)

HVORDAN PROGRAMMERES DEN?

En Pi kan programmeres på mange måder. En Pi er en normal computer, så hvis man skal lave et script der kan læse fra et keyboard og skrive ud på skærmen er der næsten ubegrænsede muligheder.

Der, hvor det bliver spændende, er når der skal programmeres hardware, når GPIO porten skal bruges.

På en Pi er en GPIO blot en fil i fil systemet som fx. `"/sys/class/gpio/gpio16/value"`

Denne fil kan man læse eller skrive til.

Man kan også bruge en af de mange biblioteker der findes. Fx. til Python eller Java. Da en Pi "bare" er en normal Linux computer kan man sagtens programmere den ved at ssh'e ind til maskinen og derefter compile sin kode. Man kan også med fx java, compile det på en anden computer og køre det på en pi efterfølgende. Dette kan gøre oversættelsen af programmet hurtigere, da en Pi ikke er hurtigere end en normal skrivebordscomputer.

I denne artikel vil der blive brugt Pi4J biblioteket til at lave et program med, men koncepterne samt opsætningen er den samme, hvis man i stedet vælger at bruge python eller C.

Til at skrive java kode kan jeg personligt anbefale IntelliJ Idea, men ethvert Java IDE kan bruges, eller endda *vim*.

Alarm!

Som eksempel til at vise programmering og opsætning af en Pi vil der blive lavet en lille simpel alarm, som man kan bruge til at overvåge og afslører folk der prøver at komme ind i ens bolig når man ikke er tilstede.

Liste af komponenter:

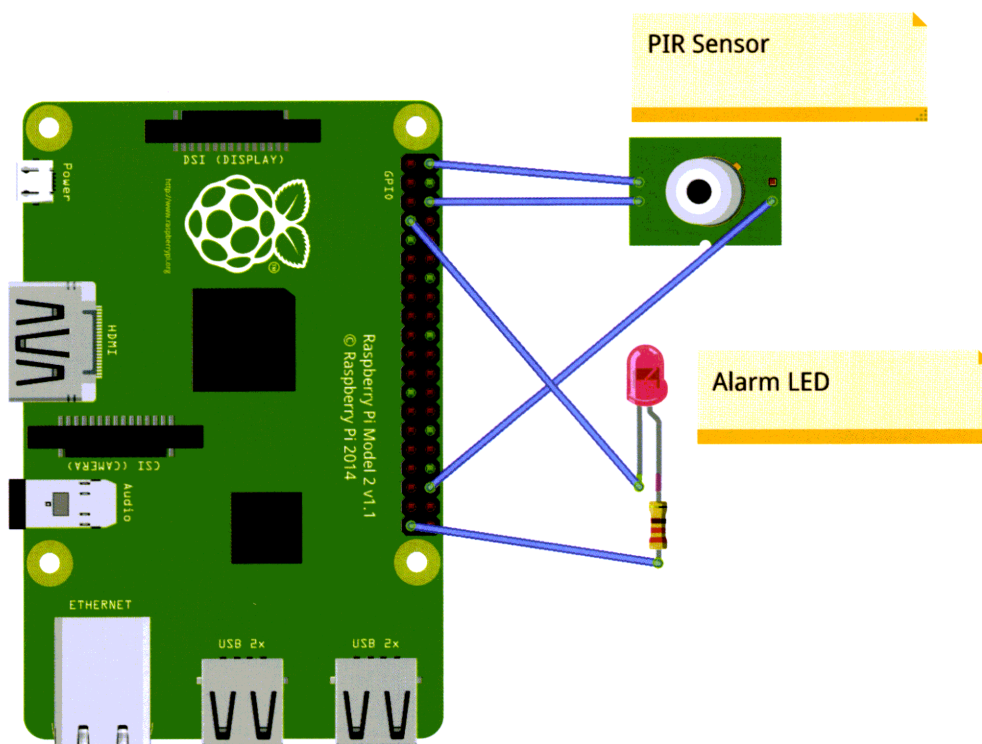
Raspberry pi 2 B (en model 1 kan også bruges med mindre modifikationer til koden)

1 x Pir sensor. Modellen der er brugt er fra dx.com

1 x Rød LED

Div. ledninger og andet til at forbinde det hele.

DIAGRAM



fritzing

Opsætning af software

For at tilgå GPIO på en pi med Java, skal man først have en adgang til GPIOen. Dette gøres på følgende måde:

```
GpioController gpio;  
gpio = GpioFactory.getInstance();
```

Her får vi en instans af en GPIO factory, som er det vi skal bruge til at "lave" vores input og output pins.

Når programmet er færdig med at køre skal der ryddes op og det gøres med følgende linje kode:

```
gpio.shutdown();
```

Dette rydder op og frigiver hvad der skal frigives.

Tænde og slukke en LED

Til at lege alarm sætter vi en LED på. Denne LED kunne være en sirene.

Først skal der laves en PIN som bliver sat op til output.

```
GpioPinDigitalOutput ledPin =  
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_07, PinState.LOW);
```

Det er det, nu er den klar til brug sammen med PIR sensoren.

Aflæsning af en pir sensor

Pir sensoren aflæses ved at markere en Pin som input:

```
GpioPinDigitalInput pin = gpio.provisionDigitalInputPin(RaspiPin.GPIO_27,  
PinPullResistance.PULL_DOWN);
```

Pir sensoren går høj, når den mærker bevægelse, derfor bruges der en pull down modstand der gør at indgangen er lav, hvis der ikke er angivet andet. Derefter skal vi sætte op så den reagerer når der er bevægelse.

I Java bliver event handlers kaldt listeneners, og derfor skal vi opsætte sådan en.

```
pin.addListener((GpioPinListenerDigital) event -> {  
    //Kode til at håndtere bevægelse  
});
```

Dette er java 8, så man kan bruge lambda notation.

Denne event bliver fyret hver gang der er en ændring. Vi er kun interreseret i når PIR sensoren går høj, så er der nemlig bevægelse. Når der er bevægelse skal der tænde en LED som viser når der er bevægelse. På det event object der kommer ind kan man spørge på om tilstanden er høj. Hvis den er høj tænder vi for LED'en.

Derfor er koden til at håndtere bevægelse følgende:

```
if (event.getState().isHigh()) {  
    ledPin.high();  
    } else {  
        ledPin.low();  
    }
```

Her tænder vi for LED'en når der bevægelse, og når bevægelsen er væk, så slukker vi.

Dette giver en samlet kode der ser sådan ud:




```

package dk.davsebamse.alarm;

import com.pi4j.io.gpio.*;

public class Alarm {
    public static void main(String args[]) {
        GpioController gpio;
        gpio = GpioFactory.getInstance();

        GpioPinDigitalOutput ledPin =
            gpio.provisionDigitalOutputPin(RaspiPin.GPIO_07, PinState.LOW);

        GpioPinDigitalInput pin =
            gpio.provisionDigitalInputPin(RaspiPin.GPIO_27, PinPullResistance.PULL_DOWN);
        pin.addListener((GpioPinListenerDigital) event -> {
            if (event.getState().isHigh()) {
                ledPin.high();
            } else {
                ledPin.low();
            }
        });
        for (;;) {
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                break;
            }
        }
        gpio.shutdown();
    }
}

```

Den sidste **for-løkke** (try - catch InterruptedException) er for at holde programmet kørende indtil vi trykker CTRL+c.

Raspberry Pi 2 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART) 15
	Ground	9		10	GPIO 16 RxD (UART) 16
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0 1
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4 4
	3.3 VDC Power	17		18	GPIO 5 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI) 10
	Ground	25		26	GPIO 11 CE1 (SPI) 11
	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0 26
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN 28
	Ground	39		40	GPIO 29 PCM_DOUT 29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Det er muligt at programmere en Pi på mange måder. Her var det Java 8 som blev brugt. Sammen med Pi4J giver det en nem måde at bruge det java programmering man måske kender i forvejen og bruge til at styre en alarm, en robot, en vandingstank, en guldfisketæller eller hvad man nu kunne finde på.

Henvisninger:

Pi4J – pi4j.com

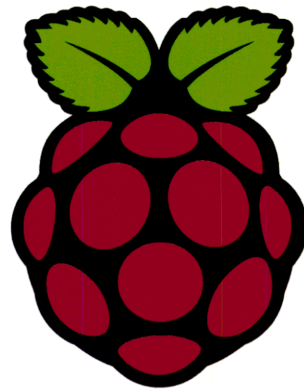
Raspberry pi foundation - raspberrypi.org

Forretninger der handler med pi:

modmypi.com

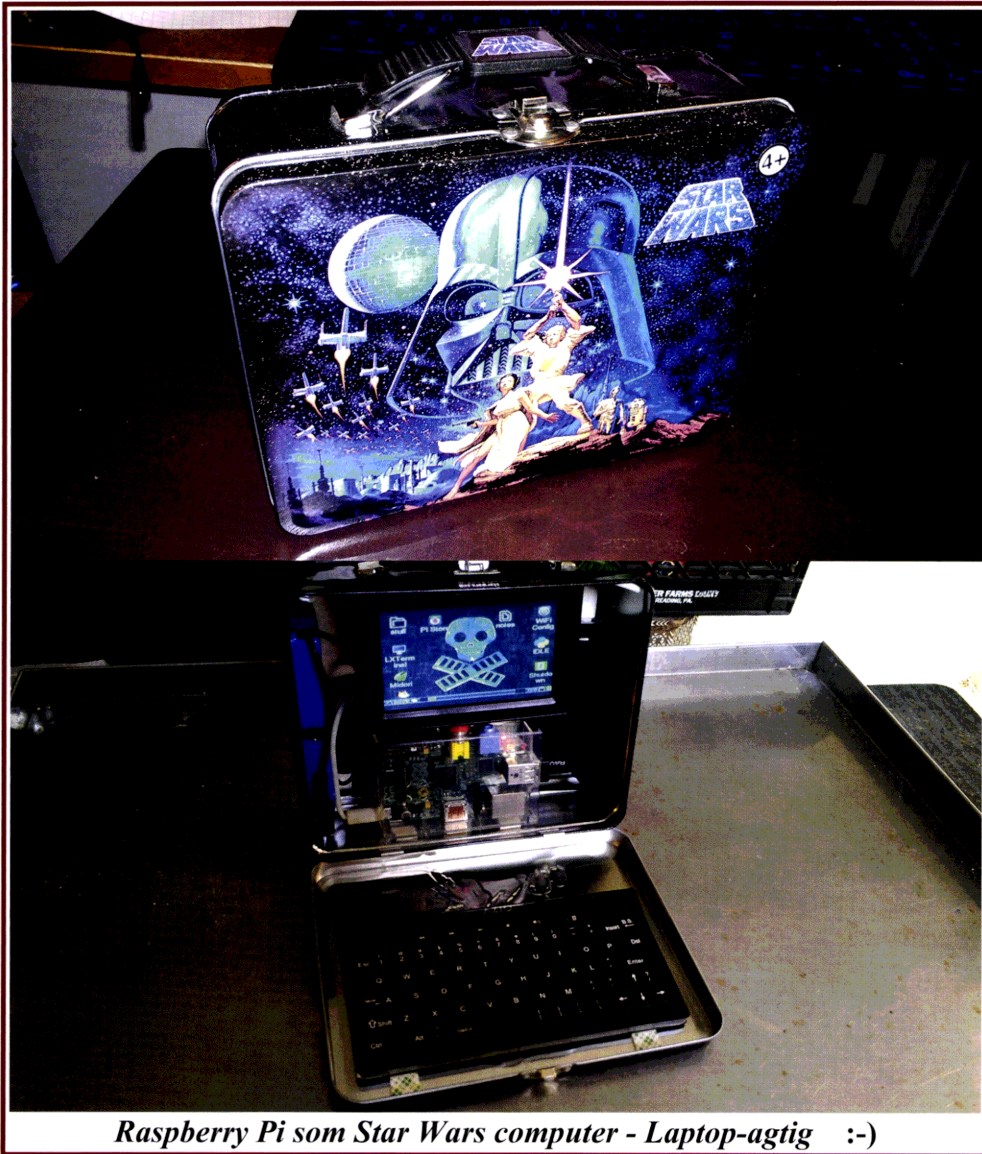
raspberrypi.dk

pir sensor fra dx - <http://www.dx.com/p/pyroelectric-infrared-pir-motion-sensor-detector-module-157004>



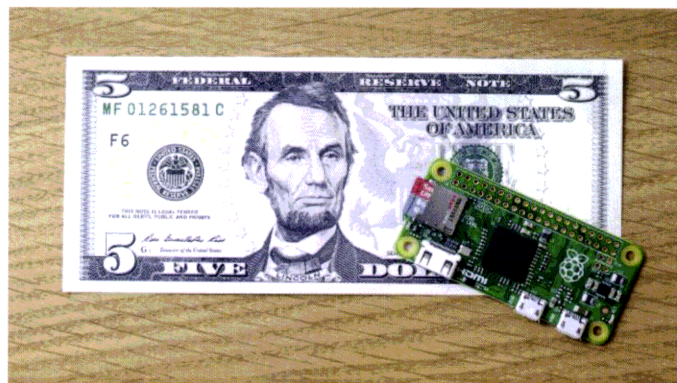
*"If Java had true garbage collection, most programs would delete themselves upon execution."
(Robert Sewell)*

(Linkedin har 197 hits på Robert Sewell!)



Raspberry Pi som Star Wars computer - Laptop-agtig :-)

En Raspberry Pi Joke-maskine:



FOSDEM - Januar 2016

Bruxelles

FOSDEM er en fri - og gratis - event for softwareudviklere, hvor man kan mødes og dele ideer og samarbejde

Hvert år samles tusinder af udviklere af fri og open source software (FOSS) fra hele verden ved denne event i Bruxelles.

FOSDEM 2016 havde officielt 5000 deltagere, uofficielt måske noget mere, eftersom man hellere vil sige for lavt et tal.

Det er ikke mere end en uge siden at det blev afviklet, FOSDEM 2015 var sidste weekend lø-sø d. 30. og 31. Januar, og allerede nu ligger der mange videoer.

De fleste af emnerne er centrale for den fortsatte trivsel af Open Source. FOSDEM har et erklæret formål: at være til glæde for Open Source udviklere og skriver selv nogenlunde sådan her:

Hvad er FOSDEM?

Free Open Source Developers' European Meeting!

FOSDEM er en fri/gratis og ikke kommerciel event organiseret af communityet for communityet. Målet er at give et sted at mødes for Free Open Source Software udviklere (FOSS developers) -

- *for at få kontakt med andre udviklere og andre projekter,*
- *for at få noget at vide om de nyeste strømninger i free software miljøet, - og i open source miljøet,*
- *for at høre interessante foredrag og præsentationer om forskellige emner fra projektledere og projektmedlemmer,*
- *for at fremme udvikling af - og fordele ved - fri og Open Source løsninger.*

Deltagelse er helt gratis, selv om organisatorerne gerne modtager donationer og sponsorater.

Developer rooms

FOSDEM teamet mener det er vigtigt at udviklere af Fri og Open Source software at mødes in persona *in real life*.

Derfor har vi opsat udvikler-rum, devrooms med netværk/internet forbindelse, projektorer, hvor man kan vise præsentationer - eller bare screendumps fra projekterne. Devrooms er stedet hvor man diskuterer, programmerer, og offentliggør sidste nye retninger, lyn-foredrag og nyheder. Vi tror at udviklere har stor glæde af disse møder.

En smule historie

Raphael Bauduin - en fan af Linux-bevægelsen i Belgien, - besluttede i år 2000 at organisere et lille møde for udviklere af Open Source software. Han kaldte det Open Source Developers' European Meeting, OSDEM.

Raphael lavede også en mailing liste, et lille website og spredte ordet til folk rundt om; blot et par uger efter, var masser af mennesker klar til en spændende event i Bruxelles!

Invitationer blev sendt til kendte personer i communityet, Rasterman, Fyodor, Jeremy Allison og så videre. De gav alle positivt respons og så var succesen hjemme.

Det andet år blev OSDEM omdøbt FOSDEM. Og nu - mange år efter, er det vokset del den event det er i dag. Hvert år prøver man at dække et spektrum af FOSS, og hvert år er der mere end 5000 deltagere - da man ikke registrerer, kan man ikke få et nøjagtigt tal, så det er et lidt konservativt sjus.

FOSDEM blev afholdt på ULB Solbosch Campus, Université Libre de Bruxelles (Det Frie Universitet i Bruxelles).

Raphael er ikke længere den drivende kraft i FOSDEM, efter 7 års hårdt arbejde trængte han til at lave noget andet, og har gang i andre Open Source projekter. Men der 22 andre, som har taget ansvaret og desuden en mindre hær af frivillige hjælpere!

Foredragsrækker

Emnerne dækker områderne fra distributioner, over enterprise (stordriftsinstallationer) og hardware til kommunikation, office software, systemadministration og virtualisering. Desuden er der korte foredrag, Bird of Feathers.

Selvfølgerig griber emnerne ind i hinanden, for eksempel er en vigtig del af nytænkningen med hensyn til versions-helvedet at man bruger containers, sandboxing til at køre applikationer, der ikke er en del af den "solide" kerne.

Langdon White holdt et foredrag om dette emne:

Re-thinking Linux Distributions

... separate the operating system from the content

Det nye er at distributionerne langt om længe erkender, at dependencies ikke nødvendigvis løser alle stabilitetsproblemer.

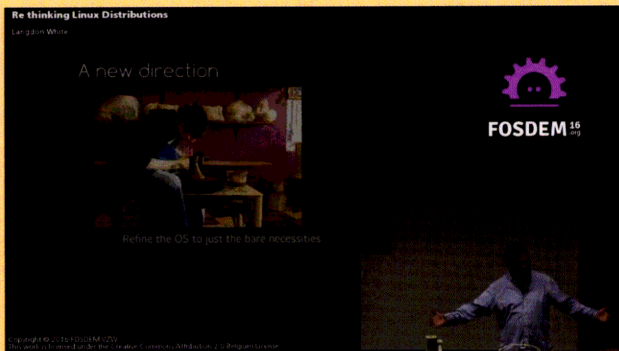
Software har forskellige levetider (eller brugstider) og software pakker er opelt i mange små pakker. Langdon nævner nyheds-websystemet *Drupal*, som er avanceret, stabilt, kan en masse, og som har brug for en webserver og en database backend.

Det er ikke altid at man bare lige kan opdatere sin database-backend. Nogle gange breaker det Drupal hvis man opdaterer backend databasen. Men tænk lige igen, Databasen er endda opdelt i flere mindre pakker.

Langdon nævner flere eksempler på enorme dependency-træer - jeg kan ikke huske dem alle, men jeg kan huske at der er pakker, som kræver hele webserveren installeret, og et andet eksempel er mailservoren, som af nogle programmer bruges til at give besked om fejl. Men man kunne jo nøjes med en attrap, det ødelægger ikke programmets funktionalitet.

I praksis har fx. Debian stable systemer kørt efter *best practice* princippet: Folk vidste, hvad der virkede. Langdon viste denne slide, som er et screenshot af Fedora download webpage (Langdon arbejder for Red Hat)

Et eksempel på at man målretter en pakkedisse til henholdsvis desktop (eller laptop), serverbrug, typisk uden grafiske programmer, og en cloud server.



Egentlig er der jo ikke noget nyt i at lave en anbefalet pakkediste til en maskine med et bestemt formål; men Langdon var ude med mere nuancerede budskaber: Man burde stræbe efter en stabil "kerne" af applikationer og de, der ikke var så stabile eller alment kendte og udbredte kunne foreslås lægges i et COPR arkiv, *Cool Other Project Repository*. Fedora distributionen giver brugerne et letanvendeligt build-system, så de kan danne et eget repository, og - efterhånden som det bliver stabilt - dele det med andre.

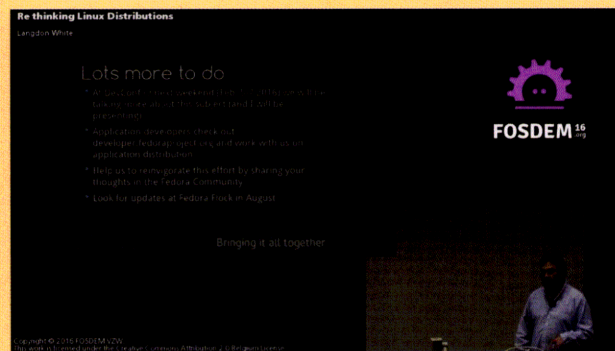
Implementation

- RoleKit: Provides a method for simplifying the distribution & installation of multi-component server software
- xdg-app: A container-izing technology to package and deliver (desktop) applications with little access to the host OS
- Atomic Workstation: An effort to use os-tree to deliver a user's workstation & provide apps separately
- Nucleule & AtomicApp: A specification and reference implementation for delivering multi-component applications in containers

Delivering self-contained modules & apps

Meningen med xdg-app er at ustabile applikationer (incl. webservices) kan køres i containere, sandboxing. Man kan lukke helt af, således at den knap så tillidsvækkende applikation går igennem xdg-app; hver gang applikationen ønsker at røre ved eller lave om på kernefunktionaliteten, skal brugeren (systemadministratoren) godkende det. Xdg-app vil spørge brugeren om tilladelse før der gives adgang.

Det kaldes application virtualisering, og det er i nogen grad inspireret af *app-store* modellen i Android og iOS. **J**



Langdon White har en liste over de ting, der bør gøres for at få mere stabile distributioner med mulighed for tilføjelse af nye, mindre stabile pakker.

xdg-app bruger OSTree til at danne bootbare, urørlige filsystemer. Det er ikke et pakkesystem og det er ikke et disk-imaging system, men noget midt imellem.

Man bruger sit yndlings-build system til at bygge og installere på en build server, og så kan OSTree via http lægge indholdet på en repository server. På hver klient kan *ostree admin upgrade* foretage en incremental opgradering.

Det skal lige suppleres at man tidligere har gjort det samme ved at bruge *rsync*. Problemet med den slags "mastersystem" er, at man ikke kan duplikere /etc (som fx. indeholder hostnavn og andre hostspecifikke konfigurationer.) Det må være OSTree opgaven at løse de problemer på en minimal arbejdskrævende måde. Det kunne minde om puppet og andre config management systemer, og strengt taget kunne det også minde om den metode, hvor man danner en pakkeliste (med apt eller lign) og så bruge den som input til pakke-installeren. Forskellen er selvfølgelig, at man så får distributionens pakker og ikke lokale ændringer; det vil man derimod få hvis man bruger OSTree, som så må håndtere /etc og /var på særlige måder.

Andre operativ-relaterede foredrag på FOSDEM

Systemadministratorer vil nyde disse foredrag (som kan downloades gratis):

- Rearchitecting Linux IO Towards Petascale Storage.
- Systemd And Where We Want To Take The Linux Userspace.
- How To Design A Linux Kernel API

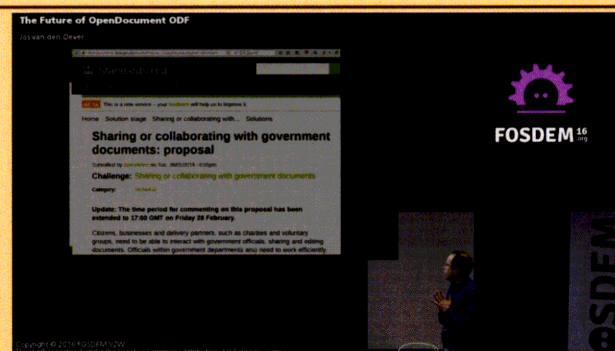
Link til FOSDEM video-arkiv:

- <http://video.fosdem.org/2016/>
- <http://video.fosdem.org/2016/janson/>

Fremtiden for Open Document Format

En statsansat i den hollandske regering fortalte om brugen af Libre Office i administrationen i Holland, hvor man har sat sig det mål at al kommunikation med borgerne sker digitalt fra 2017. Han havde også et sideblik til UK, som har haft en høring om hvilket dokumentformat, der burde bruges i fremtiden, og hvor et flertal pegede på ODF.

Som bekendt har MSOffice også nogen support for ODF, og som bekendt er softwaren ikke uden problemer, men ideen bag, at det skal være et åbent, ikke kommercielt format, er jo indlysende. Der er ikke nogen sanktioner, men kun anbefalinger, og det er nok meget godt.



Her taler Jos van den Oever om fremtiden for ODF, maintaining the momentum -og viser UK regeringens proposal - man kan som nævnt finde det på video.fosdem.org.

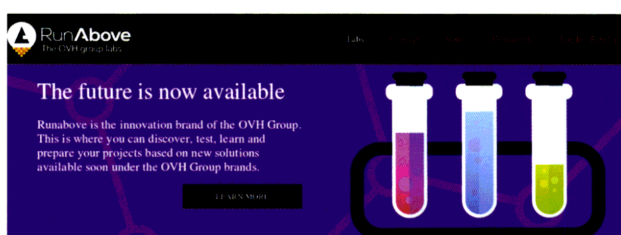
"Computers are useless. They can only give you answers." (Pablo Picasso)

Fortsat fra side 2 ... (Foredrag på OSD)

Et andet foredrag, som i skrivende stund er sat på programmet, er *Everything You Always Wanted to Know About MariaDB But Were Afraid to Ask* - et foredrag af Anders Karlsson.

Anders Karlsson er konsulent (programmør), salgssingeniør, og (surprise ...) foredragsholder med lang erfaring i database software. Han har arbejdet for flere af de større database software firmaer - og derfor kan man vel håbe på, at han fortæller den fulde historie om MariaDB og hvordan den forholder sig til MySQL, der som bekendt røg ud af de oprindelige udvikleres kontrol og blev solgt til SUN microsystems. Året efter blev SUN så i sin tur solgt til Oracle.

Desuden vil Karlsson fortælle lidt om MariaDB på IBM Power8 platform. Power8 er en familie af superskalare symmetriske multirprocessorer baseret på *Power - Performance Optimization With Enhanced RISC*. IBM supporterer Linux på Power8 - der er en model L, som kun kører Linux. Nogle webservices har taget Power8 i brug, en af dem, som Karlsson nævner på MariaDBs hjemmeside, er runabove.com - en service som vil hjælpe entreprenante kreative mennesker med at realisere ideer.



All Labs



RunAbove kører ideen igennem en række "labs" så den ender som et produkt - forhåbentlig! RunAbove bruger også MariaDB på Power8

Det må også nævnes, at Michael Widenius, grundlægger af MySQL og også MariaDB, holder keynote. Man kunne derfor savne at der var et foredrag om PostgreSQL, som - trods den mindre markedsandel - længe har været på forkant.

Der er mange flere emner og foredrag allerede nu i skrivende stund, for et musik+data menneske som mig er Niklas Nisbeth om Axoloti, Open Source musik-hardware og software, særligt spændende. Troels Arvin om monitorering med Check_MK, og meget mere kan man læse om i foramtalen af OSD andetsteds i bladet.

Open Source udbredelse

Det er vanskeligt at se, hvor stor tilslutning de enkelte OSS projekter har, indtil de har eksisteret i nogle måneder eller år, og de fleste af os har bare brug for at vide, om de er med i en stabil distribution af Linux (eller BSD).

Men det er måske karakteristisk for events i det tekniske miljø at der er mere vægt på projekter end der er på IT-politik og på status for Open Source.

Vil det fx. være muligt at sammenligne omkostningerne for en databaseregistrering af en speciel sygdom til forskningsmæssig ved at opsætte en webserver med Open Source og med backup og forholdsvis billig hardware? Mine erfaringer fra RUC er, at det er endog meget muligt og forholdsvis nemt. Mere vanskelig er sikkerhed og trådløst net - det foregår ofte ved forskellige

avancerede teknikker baseret på hvad man kan med Cisco routere.

I tidligere numre af DNYt har vi været inde på at Open Source er mere accepteret i andre lande. Danmark kaldes somme tider et MS-land.

De mest fremtrædende use cases for Open Source er Münchens LiMux projekt, som begyndte i de tidlige 0'ere med at bystyret valgte at kassere Microsoft produkter fordi et skift og opdateringskrav viste, at de var udsat for teknisk indlåsning, technical lock-in.

Lock-in er, når man er bundet til en bestemt leverandør, for hvis man bruger andet, bortfalder garantien, eller man kan ikke styre adgang og sikkerhed mv.

München er en af de eneste socialistiske byer i Bayern, og her var man forstående overfor IT-ledelsens klager over at være bundet på hænder og fødder. Måske har det hjulpet at se på regnskabet for opdatering af 14000 PC'er fra MS-NT til XP fordi supporten ophørte.

Efter 10 år havde München sin egen LiMux, en Ubuntu baseret distribution (af Linux) og 15.000 PC'er og laptops, som kører Open Source software.

Endnu mere udbredt er Open Source i spansk-talende lande, Spanien har et institut for overvågning af Open Source udbredelse, som er kommet frem til at 9 ud af 10 servere bruger Open Source, mens 8 ud af 10 desktoppe i det offentlige bruger Open Source desktop software.

Desuden er der en *Open* politik i Baskerlandet hvor al software, som produceres, i princippet skal være Open - altså tilgængelig for andre at se hvad der er skrevet og med mulighed for at genbruge det.

Men det er som om selve diskussionen om fordele og ulemper - fejl og stabilitet - er gået i glemmebogen. Eric Raymond prøvede med essay om Open Source principper at beskrive, hvad det er, der gør at et OS projekt har succes.

Raymonds analyse omfatter både økonomien og arbejdsdelingen. Han siger at Open Source udviklere har gennemhullet det tidligere så anerkendte bemandings-princip, beskrevet af Frederick Brooks i bogen *The Mythical Man Month*, nemlig at det ikke nytter at sætte flere programmører på ét projekt, fordi de skal oplæres og det tager lang tid.

Men samtidig kommer han også ind på at der er en eller ganske få udviklere, som bærer ansvaret. Hele "fidusen" er at man deler arbejdet på en måde, så det er muligt at teste kode-bidrag uden at projektet bliver brudt - broken code. For Linux kernens vedkommende er der et hierarki af udviklere, som godkender kode, og først, når de har set på det, og kommenteret, kommer det til Linus Torvalds, der stadig har det afgørende ord.

Der er morsomme sager, hvor Torvalds har kasseret kode, fordi man ikke kan læse 300k kode uden kommentarer - men hvor andre har sagt, at denne klump faktisk fungerer bedre end andre bidrag, som skal udføre de samme funktioner. Men Torvalds stod fast på at udokumenteret ulæselig kode ikke fik adgang - og så måtte den nedbrydes i mindre klumper.

Er du stadig i tvivl om hvorfor Open Source er vigtigt?

Som et sidste eksempel er her en oplevelse fra en søgning på en bogtitel:

Krav: For at se denne bog skal du have en iPad med iBooks 2 eller nyere og iOS 5 eller nyere, eller en iPhone med iOS 8.4 eller nyere, eller en Mac med OS X 10.9 eller nyere.

Donald Axel

Kommando(linie)centralen

GNU GPG underrubrik

Denne klumme introducerer i hvert blad sjove og (u)nyttige programmer, som kan køres fra en kommandolinie - en intro til programmører om hvordan computeren fungerer. (I parentes bemærket kan desktoppen, grafikken med login og det hele, også køre fra en kommandolinie.)

Kommandolinien er stort set en wrapper for `execve(2)` - (execute med vector også kaldet argumenter og environment.) systemfunktionen, som starter et andet program. Når man klikker på et ikon, er det en kommandolinie, man aktiverer, og man kan selv konfigurere sådan en kommandolinier, hvis man mangler en funktion på sin desktop.

Send aldrig personnummer, bankkonto eller andre oplysninger i mail. For øvelsens skyld, så lad os bare sige at vi har en god grund til at kryptere noget, hvordan gør vi så?

```
gpg --gen-key
```

Man får en bunke spørgsmål - men programmet finder selv ud af hvor man skal gemme krypteringsnøgler (~/.gnupg)

```
gpg --list-keys
```

Kryptering bruges også til at danne en signatur - en art checksum på en fil, som man kan kontrollere at filer ikke er ændret undervejs eller over tid. Begge ting forudsætter at de nøgler man bruger, er sikre.

Mekanikken i denne form for kryptering er private/public key kryptering. Fx. krypterer man med sin private key, offentliggør public key ved at sende den til fx. keys.gnupg.net, hvorefter man sender sin krypterede mail som attachment.

Modtager af den krypterede fil / mail kan dekryptere den med vores public key.

Demonstration:

```
$ date +"Klokken er %T - PID er $$" > bla.txt
$ cat bla.txt
Klokken er 23.32.01 - PID er 13245
$ gpg --encrypt bla.txt \
  --recipient "Donald Axel"
$ # eller med korte options:
$ gpg -e -r "Donald Axel" bla.txt
$ ls -lo
total 8
-rw-r--r-- 1 toot 35 Feb 7 23:32 bla.txt
-rw-r--r-- 1 toot 372 Feb 7 23:35 bla.txt.gpg
$ cat -v bla.txt.gpg
```

Skærmen fyldes af uformateret underligt krimskrams, cat betyder som bekendt concatenate, -v betyder vis kontrol-bytes som printbare tegn.

```
$ echo "Nu modtager vi denne fil i mail :)"
Nu modtager vi denne fil i mail :)
$ gpg --output laes.txt --decrypt bla.txt.gpg
You need a passphrase to unlock the
secret key for user: "Donald Axel (Retired)":
```

Vi indtaster det hemmelige password, som vi brugte ved `gpg --gen-key`.

```
$ cat laes.txt
Klokken er 23.32.01 - PID er 13245
$
```

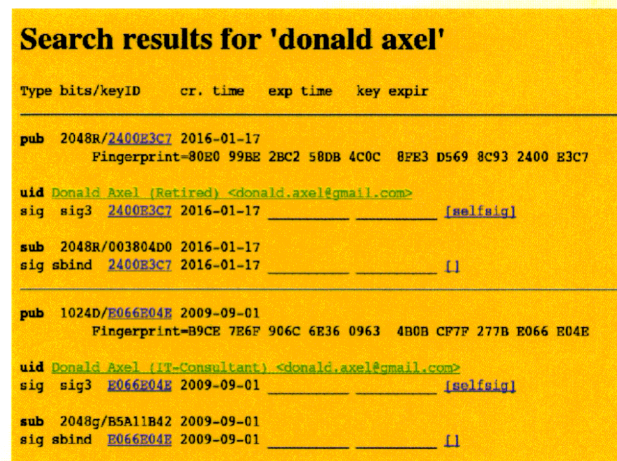
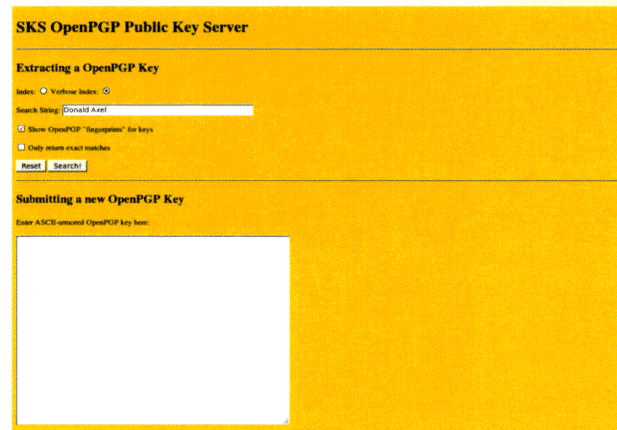
Sådan.

Men det er jo ikke særlig nyttigt. Kan vi sende den som mail fra et script? Ja med `sendmail` klienten kan man sende mail med attachment og alting. Derom i næste nr af DNYt! Men modtageren skal jo så have vores public key for at kunne afkryptere?

Det er hele humlen og også det, som gør det lidt træls at bruge denne kryptering. Men heldigvis kan `gpg` klare det fra et script (program).

Først må vi dog lige vise en keystore, `keys.gnupg.net`

Først tager vi et kig via en browser:



Men som sagt kan man godt hente en key ved hjælp af et script med kommandoen:

```
$ gpg --search-keys "Ib Olsen"
gpg: searching for "Ib Olsen" from \
hkp server keys.gnupg.net
(1)   Ib Olsen (Revisor) <ibo@gmail.com>
//osv.osv - der kan være rigtig mange.
```

Men det er skam ikke kun en søgning, vi får lov at indtaste et nummer og den key bliver så importeret (hvis den ikke er i vores nøglering i forvejen.)

Iøvrig skal vi selvfølgelig også lige lægge vores key op:

```
$ gpg --armor --output pubkey.txt \
  --export 'Donald Axel'
```

Kig på filen med fx. `cat` - og denne fil kan sendes til keys serveren via web-interfacet (se ovf.)

Men vi kan også sende den fra kommandolinien - og dermed kan funktionen `scripts`, som man siger.

```
$ gpg --send-keys 2400E3C7 --keyserver
```

Minsandt! `gpg` fandt ud af at sende til keys.gnupg.net.

De fleste mailprogrammer til Linux kan bruge `gpg`.

"Never trust a computer you can't throw out a window." (Steve Wozniak)

Internet of Things



SuperUsers nye IoT-kurser:

IoT Overblik:

På vores 1-dags kursus får du det samlede overblik over anvendelser, platforme og teknologier indenfor IoT.

IoT Programmering:

Kom på et af vores 3-dages programmeringskurser og lav din egen IoT enhed. Her bliver du både udstyret med viden og hardware til selv at kunne lave IoT-enheder.

IoT Sikkerhed:

Når alle ting pludselig kan tilgås fra internettet, er der også nogle sikkerhedsmæssige udfordringer. På vores 2-dages kurser lærer du at udforme sikre IoT-løsninger.

Windows 10 IoT



Linux IoT



Microcontroller IoT

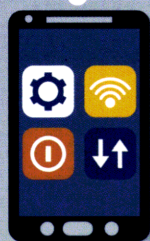
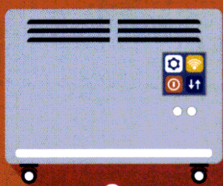
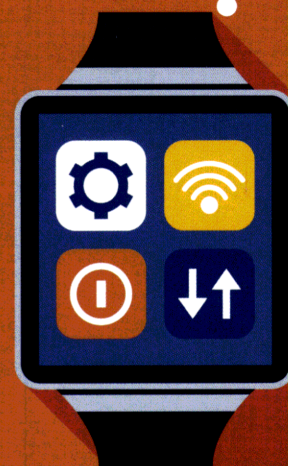


SuperUsers nye IoT-kurser

Se mere på: www.superusers.dk/iot



IoT - Internet Of Things - er den nye bevægelse, hvor alle tænkelige enheder bliver koblet på internettet. IoT kan være alt fra industrielle sensorer/transmittere til enheder i det private hjem, som med forbindelse til internettet får nye anvendelsesmuligheder. IoT er i ekstrem vækst. I dag er der 7 mia. enheder knyttet til internettet!



SUPERUSERS

Danmarks største IT-kursushus
Afdelinger i Aarhus og Hillerød