

- ▶ Jimmy Hedman
- ▶ South Pole AB
- ▶ Linuxkonsult sedan -00
- ▶ Fokus på beräkningskluster



Upstart (ersätter /sbin/init)

- ▶ Filosofi
- ▶ Konfigurationsfiler
- ▶ Debugga
- ▶ "Demo"



Filosofi

- ▶ Maxa CPU
 - ▶ Utnyttja multicore maximalt
- ▶ Maxa IO
 - ▶ Varför vänta på disk när andra saker kan köra samtidigt?
- ▶ Signal-baserat
 - ▶ Startar ALLT, men saker kan vänta på andra signaler.



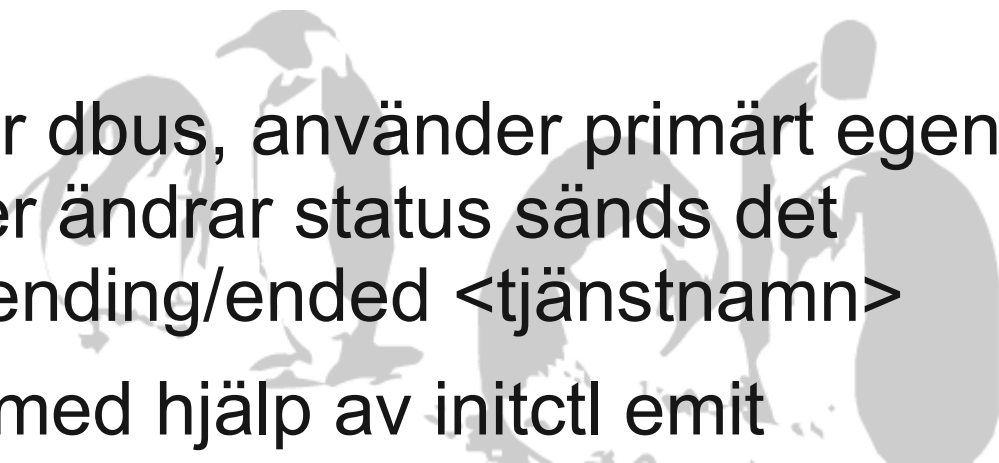
Filosofi

▶ Parallellt

- ▶ Allt som inte är beroende av något går igång parallellt
- ▶ Allt eftersom förväntade signaler kommer går väntande tjänster igång

▶ "Signaler"

- ▶ Kan signalera över dbus, använder primärt egen "buss". När saker ändrar status sänds det starting/started/ending/ended <tjänstnamn>
- ▶ Godtycklig signal med hjälp av initctl emit <signal>



Filosofi


- ▶ Ingen pid-fil
- ▶ ptrace för att "följa" processerna



Konfigurationsfiler

- ▶ description, author, version, emits
 - ▶ dokumentation
- ▶ start on, stop on
 - ▶ väntar på signal (ej state), kan vara booleanska uttryck (or, **and**) och reguljära uttryck (runlevel [23])
 - ▶ gdm.conf:

```
start on (filesystem
          and started hal
          and tty-device-added
          KERNEL=tty7
          and (graphics-device-
added or
udevtrigger) )
stopped
```



Konfigurationsfiler

- ▶ **expect** `stop | daemon | fork`
 - ▶ När är tjänsten igång. 'stop' väntar på en STOP (svarar med en CONT), 'fork' väntar på en fork och 'daemon' väntar på en dubbel fork.
- ▶ **exec** `kommando [argument]`
 - ▶ Kör kommandot (med argumenten)
- ▶ **script ... end script**
 - ▶ shell-script för att starta
- ▶ **pre-start/post-stop** `<exec|script ... end script>`
- ▶ **pre-stop/post-start** `<exec|script ... end script>`



Konfigurationsfiler

- ▶ `console` `output | owner`
 - ▶ Default går `stdin`, `stdout` och `stderr` till `/dev/null`. Med `console output` så går dom till `/dev/console` och med `console owner` så blir även `/dev/console` ägare av processen (tar mot signaler tex `ctrl-c`).
- ▶ `respawn [limit COUNT INTERVAL]/task`
 - ▶ Är det en "service" (`respawn`) eller en enstaka händelse (`task`). Kan även begränsa hur snabbt den ska respawn'a.
- ▶ `kill timeout TIMEOUT`
 - ▶ specificerar tiden mellan `SIGTERM` och `SIGKILL`.

Konfigurationsfiler

- ▶ **env** `KEY=VALUE`
 - ▶ definierar miljövariabel.
- ▶ **export** `KEY`
 - ▶ exporterar miljövariabel till signalen.
- ▶ **instance** `NAME`
 - ▶ Tillåt flera instanser av scriptet men med olika "namn", tex eth0 eller eth1.
- ▶ **normal exit** `signal|exitcode`
 - ▶ Vilka signaler och exit-koder räknas som normalt avslut

Konfigurationsfiler

- ▶ **umask** `UMASK`
- ▶ **nice** `NICE`
- ▶ **oom** `ADJUSTMENT | never`
 - ▶ justerar hur "känslig" oom är för denna process.
- ▶ **chroot** `DIR`
- ▶ **chdir** `DIR`
- ▶ **limit** `LIMIT SOFT | unlimited HARD | unlimited`
 - ▶ sätter "ulimit" för processen. `LIMIT` kan vara `core`, `cpu`, `data`, `fsize`, `memlock`, `msgqueue`, `nice`, `nofile`, `nproc`, `rss`, `rtprio`, `sigpending` eller `stack`.

Använda

- ▶ start/stop/restart <tjänst>
 - ▶ motsvarar /etc/init.d/<tjänst> start|stop|restart
- ▶ status
 - ▶ Kollar status på tjänstern, returnerar om den är igång och vilken pid den har.



Debugga

- ▶ `--debug` till kärnan (ta bort quiet och splash)
- ▶ `initctl emit <signal>`
- ▶ `initctl list`
- ▶ `(bootchart)`



Styrkor och svagheter

- ▶ Startar snabbare
- ▶ Utnyttjar multicore
- ▶ Utnyttjar diskar
- ▶ Komplex
 - ▶ ptrace
 - ▶ besvärligt felsöka



Vilka använder upstart?

- ▶ Fedora ≥ 9
 - ▶ "compatible mode" i F12, kanske i F13.
- ▶ Ubuntu ≥ 6.10
 - ▶ Först i 9.10 började man konvertera från "compatible mode".
- ▶ Maemo 5
- ▶ Palm Pre's webOS
- ▶ Chrome OS



Demo

- ▶ Boota Karmic med –debug
- ▶ initctl list
- ▶ Visa tty1.conf
- ▶ Visa gdm.conf
- ▶ Visa mountall.conf

