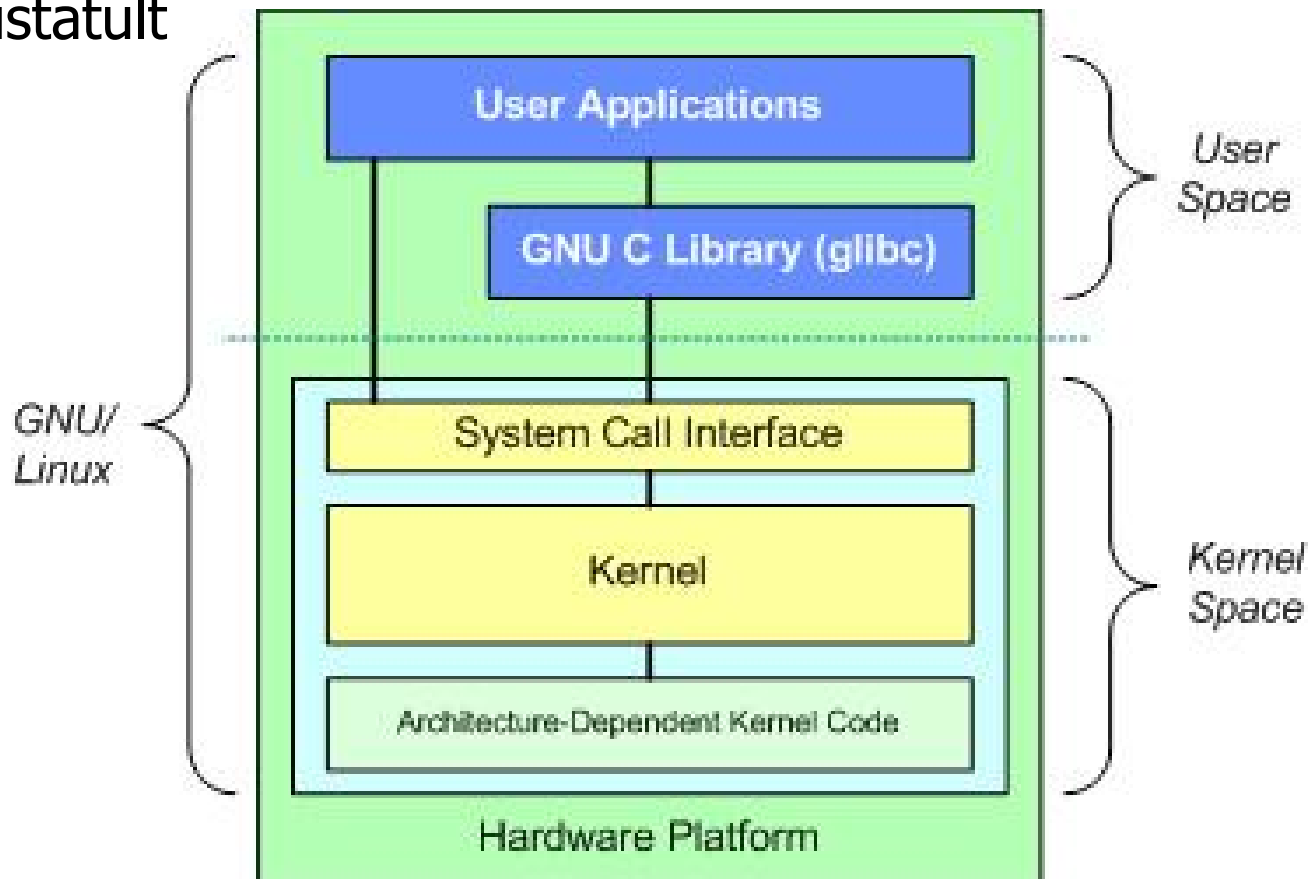


Protsesside, mälu jm haldamine



Linuxi ehitus (struktuur)

Lihtsustatult



Protsess



- Multitasking - palju protsesse töötab paralleelselt
- Tuumas asub protsesside tabel
 - igal protsessil on identifikaator PID
 - igal protsessil on vanem, kes ta käivitas PPID
 - iga protsessiga on seotud kasutaja UID ja grupp GID
- Igal protsessil on oma eraldatud ala mälus
 - virtuaalmälu
- Protseesse loovad teised protsessid *fork()* ja *exec()* süsteemifunktsioonide abil

Koormuse jälgimine



- *top* - protsessori ja mälu kasutus
 - näitab töötavaid programme tabelina
 - võimaldab tabelit erinevalt sorteerida
 - näitab mälu ja protsessorikasutust
- *vmstat* - virtuaalmälu ja ketta statistika
 - näitab süsteemi tööd iseloomustavaid parameetreid
 - argumendiks võib anda ajavahemiku sekundites, mille kohta saab regulaarset statistikat
- *iostat* - cpu ja ketta statistika
 - argumendiks võib anda ajavahemiku sekundites, mille kohta saab regulaarset statistikat
 - sisaldub pakis *sysstat*

Protsesside jälgimine

■ Ülevaate saamine - *ps, top, w, pstree*

- `ps ef` kõik protsessid koos PPID-ga
- `ps ax` kõik protsessid
- `ps axf` kõik protsessid protsessipuuna
- `ps axl` rohkem andmeid protsesside kohta
- `ps axu` protsesside kasutajainfo
- `ps axuf | grep juku` Juku protsessid puuna

■ Proovi!

```
ps ax | wc -l
```

```
ls /proc | grep \[0-9\] | wc -l
```

```
ps aux
```

```
ps auxww
```

***/proc* failisüsteem**



- Igal protsessil on oma kataloog */proc* all
- */proc* failisüsteemis on veel mitmesugust infot:

cpuinfo	info protsessorite kohta
devices	draiverite nimekiri
interrupts	kasutatud katkestused
loadavg	keskmine koormus
mounts	ühendatud failisüsteemid
stat	statistikat süsteemi kohta
sys/	mitmed tuuma parameetrid
version	Linuxi tuuma versioon

Signaalid, protsesside “tapmine”

- Signaalid (vaata *signal(7)*) (man 7 signal või kill -l)
SIGHUP(1), SIGTERM(15), SIGKILL(9)
- Protsesside „tapmine”, signaalid - *kill*
 - kill -1 PID - konfiguratsiooni uuendus, mõnel puhul viisakas lõpetamine
 - kill PID - protsessi viisakas lõpetamine
 - kill -9 PID - protsessi sundlõpetamine
 - killall protsessinimi - kõikide samanimeliste lõpetamine
 - pkill -u kasutajanimi - kõikide kasutaja protsesside lõpetamine
 - pgrep -u kasutajanimi - kasutaja protsesside nimekirja vaatamine
- Signaale saab saata
 - tavakasutaja ainult enda protsessidele
 - juurkasutaja kõigile protsessidele

Protsesside prioriteedid (*nice, renice*)



- Tuum määrab protsessi tegeliku (CPU) prioriteedi mitmete tegurite koosmõjul, milledest üks on kasutaja muudetav „niceness” tase
- Suurem „niceness” tase – kehvem prioriteet
- Tavaline kasutaja saab määrata taseme 0 .. 19 ning seda ainult suurendada
- Juurkasutaja saab määrata ka taseme -20 .. 19 ning seda suurendada ja vähendada

```
nice -n tase käivitata_programm argumendid
```

```
renice -n tase PID
```


Protsesside sidumine konkreetse CPU-ga (*affinity*)



■ Utiliit taskset

`taskset mask käivitata_programm argumendid`

`taskset 3 ls` - seob `ls` käsu CPU-dega 0 ja 1

Protsessi I/O prioriteet (*ionice*)

- *ionice* käsk määrab protsessi sisendi-väljundi klassi ja prioriteedi (ketta-I/O)
- On kolm prioriteetide klassi:
 - *Idle* (3) protsessi I/O saab ketta-aega vaid siis kui mõni teine protsess seda ei vaja
 - *Best-effort* (2) tavaline, saab määrata ka prioriteedi (0-7)
 - *Realtime* (1) kettapöörduse kõrgeima prioriteedi klass
- Tavaliselt kasutatakse klassi „Best effort” (2) ja prioriteeti 4
- Vaikimisi I/O prioriteet sõltub protsessi „niceness” tasemest ja arvutatakse valemiga $io_priority = (cpu_nice + 20) / 5$
`ionice -c 2 -n 5 -p PID`
`ionice -c 2 -n 6 käivitav_programm argumendid`

Programmide käivitamine



- Käivitamiseks peab failil olema käivitaja jaoks kehtiv x pääsuõigus (loabitt)
- Käivitataavaks programmiks võib olla
 - Kahendfail (*binary file*)
 - | Peab olema kindlas vormingus, mida tuum oskab käivitada
 - Sisaldab vastavat tunnuskoodi („*magic number*“)
 - Skript
 - | Peab esimesel real sisaldama interpretaatori asukohta

Lihtne shelli skript



`#!/bin/bash`

kohustuslik algus, interpretaator

`MUUTUJA=väärtus`

luuakse muutuja ja antakse väärtus

`korraldus argument`

täidetakse korraldus

`korraldus $MUUTUJA`

argumendiks muutuja väärtus

`korraldus $1 $2 $3`

argumendiks skripti 3 argumenti

Skript tuleb muuta täidetavaks (x pääsuõigus)!

Käivitamine tulevikus (*at*)

```
$ at käivitamisaeg
```

```
at> korraldused
```

```
at> <ctrl>d
```

```
$ echo "korraldus" | at käivitamisaeg
```

■ Käivitamisaja vormingud

- HH:MM

- Tuesday; Feb 2; Mar 3, 2002

- now; noon; tomorrow; now + 7 days

■ Tulemus saadetakse mailiga

- *atq* - näitab järjekorras olevaid töid

- *atrm* - võimaldab töid järjekorrast kustutada

Regulaarne käivitamine (*cron*)

- *cron* deemon töötab taustal ja käivitab ettenähtud ajal vajalikke programme
- programme saab ette anda *crontab* korraldusega
- mõistlik on teha endale kodukataloogi fail *.crontab* ja installeerida selles sisalduvad korraldused käsuga

```
crontab .crontab
```

- käivituste tulemused saadetakse mailiga

```
crontab -l      - kasutaja oma crontab-i vaatamine
```

```
crontab -e      - oma crontab-i redigeerimine
```

```
crontab -r      - oma crontab-i kustutamine
```

```
crontab -l -u user - teise kasutaja crontab-i vaatamine  
                  (ainult root)
```

Crontab faili vorming (kasutaja)

0,20,40 7-19,23 * * mon-fri kratsikukalt

- minut iga 20 minuti tagant
 - tund kell 7 kuni 19 ja 23
 - kuupäev sõltumata kuupäevast
 - kuu sõltumata kuust
 - nädalapäev esmaspäevast reedeni
 - korraldus korraldus koos parameetritega
-
- Korraldus täidetakse, kui kellaaeg vastab kõigile esimesel 5-l väljal olevatele tingimustele
 - Süsteemne crontab fail */etc/crontab* , */etc/cron.d/**
 - võimaldab määrata kelle õigustes käivitatakse

Cron ja at teenuse piiramine



/etc/cron.allow, /etc/cron.deny

/etc/at.allow, /etc/at.deny

- Kui *allow* fail eksisteerib, siis saavad teenust kasutada ainult need, kes selles on (iga kasutajanimi eraldi real)
- Kui *deny* fail eksisteerib, siis selles olevad kasutajad teenust kasutada ei saa

Kellaaja hoidmine (klassika)

- Ajavööndi info kirjas failis */etc/localtime*

- (Kellaaja sättimine *rdate*, *netdate*

`rdate -s time.tlu.ee`)

- Riistvaralise kella sättimine

`/sbin/clock --systohc` raua kell softi järgi

`/sbin/clock --hctosys` softi kell raua järgi

- Regulaarne sättimine (*crontab*)

`10 5 * * 0 root rdate -s mingi.kellasever.ee; /sbin/hwclock --systohc`

- Rocky Linux 8 kella parameetrite seadistamise utiliit

`timedatectl status` vt ka 'man timedatectl'

- Kellaaega võib jagada võrku ka xinetd (sisemine) teenus
time

Kellaaja hoidmine (vanemad linuxid)



- Täpse kellaaja teenus - ntp
- Arvutis jookseb ntp daemon (*ntpd*), mis sünkroniseerib kella serveri kellaga arvestades võrgus tekkivaid viivitusi jms.
- Käsitsi kellaaja sättimine
`ntpdate ntp.serveri.nimi.voi.IP`
- Vaata ka
https://kuutorvaja.eenet.ee/wiki/Arvuti_kellaaeg

Kellaaja hoidmine (CentOS8, Rocky8,9)



- Kella sünkroniseerimise teenus – chrony
- Arvutis jookseb daemon chronyd, mis samuti sünkroniseerib arvuti kella ajaserveri kellaga
- chronyc – utiliit chronyd kontrollimiseks/juhtimiseks
- Konfiguratsioon failis /etc/chrony.conf