

Unix

Kasutatud lühendid:

U	Unix
A	arvuti
S	süsteem
P	programm
D	dokument
T	tuum
F	fail
K	kataloog

1. Sissejuhatus

OS-de perekond, kirjutati 60.a. lõpus kui interaktiivne mitmetarbija S, 1. A-st sõltumatu, mitmed teoreetilised uuendused, üks enamlevinud, peamiselt keskmise võimsusega A-tel (suurtel), INTERNET, C, keerulised käsud

2. Ajalugu

60.a keskel Bell Telephones Laboratories OS *Multics* projekt (interaktiivne mitmekasutaja S), loobuti

Ken Thompson, Dennis M. Ritchie jt. PDP-7-le

KT: - luua selline OS, koordineerida grupi jõupingutusi softi loomisel, OK

KT: - kasut. U patendi tekstiD-de ettevalmistamiseks

1. U versioon 1971.a. PDP-11; seal 2 suunda (D-de ettevalmistamine, programmeerimine)

KT: OS, Ass-r PDP-7, mõned utiliidid

1973. D. Ritchie kirjutas U-i (T ja utiliidid) ümber C-s (erinevad A-d)

70-te algul akadeemilised ringkonnad, laiem levik 70-te lõpus (1977 500 U-t)

80 - U 8-bitistele mikroA

81 - U 16-bitistele mikroA

83 - U-i kommertslevik

84 - üle 10 000 U-i

U-s mõned uued ideed:

- konveier
- spetsiaalsed programmeerimisvahendid

Olulisemad Unixid ja neid tootvad firmad on:

AIX	IBM
A/UX	Apple
4.2 BSD	Berkeley Software Distribution
Coherent	Mark Williams
CPIX	IBM
DOMAIN/OS	Apollo
Genix	National Semiconductor
HP-UX	Hewlett Packard
386/IX	Interactive Systems
Mach	MT XINU Inc
OSF/1	Digital
SCO Unix	SCO
Solaris	Sun
SunOS	Sun
ULTRIX	Digital
Unix	AT&T
UTS	Amdahl
Xenix	Microsoft
Zeus	Zilloc

U. Vallner

3. U plussid

- kirjutatud C-s, ülekantav
- lähtetekstid kättesaadavad
- hea OS
- U-i P-keskkond rikas
- võimalus koostööd teha mitut P-i 1 töö tegemiseks

4. Põhimõisted

OS, käsk, takt, CU, CPU, RAM, aadress, HD, arvutussüsteem

Arvutussüsteem - A+P-varustus(OS, teenindusp-d, rakendusp-d).

Iga arvutussüsteemi oluline osa on failide organiseerimise S.

Inimene - A. 1-tel A-tel OS puudus

Ajajaotus - mitu tarbijat töötab üheaegselt, tarbijal tunne, et üksinda A-l

Multitasking (multitasking) - tarbijal mitu ülesannet

Interaktiivne S - S pidevalt valmis

T - mõningaid süst. f-e tuhandeid kordi/s; pidevalt mälu - tuum

T koordineerib A tööd, jaotab ressursse

Paljusid täidetakse aeg-ajalt (utiliidid)

U on projekteeritud nii, et min-da f-de arvu, mida täidab T

P-d. P on sisuliselt ainuke vahend inimese ja A suhtlemisel

Interaktiivse töö korral P tahab infot ja annab välja tulemused

Hea P - min infot, max tulemusi, universaalne, kitsendused

Kõik P-d 2: utiliidid, rakendusP-d

Utiliidid antakse tarbijatele koos OS-ga

Kui P ei tööta - HD-le

U-s nim. töötavat P-i **protsessiks**

Süsteemi monitor ja redaktor

Reatarbija enamiku ajast 2 P-i monitori (**shell**) ja redaktorit (editor)

Üks kõige tähtsam P U-s on **shell**, interaktiivne, tema juhtimine käskudega, mida ta interpreteerib

shell - käsuinterpretaator

shell - ka programmeerimiskeel

5. Mis on U?

P-de kogum, mis juhib A tööd, seostab tarbija A-ga, tagab tarbijale töövahendid

- kitsas mõttes ajajaotusrezhiimi OS-i T
- laias mõttes T + teised komponendid(komp-d, redaktorid, utiliidid, käsukeeled, ..)

U-i funktsioonid:

- data hooldamine failisüsteemis (5 operatsiooni) kasutusload
- P-de laadimine ja täitmine (3 operatsiooni) C jt.
- side A ja perifeeria vahel (välisseadmed - failid)
- ajajaotusS (timesharing)
- multitegurezhiim

U-i komponendid:

- T
- FS
- shell
- U-i riistad e. vahendid (Unix tools)

U-s eristatakse 3 kihti:

- T
- shell
- tarbijakeskkond
 - programmeerimiskeskond
 - tekstitöötlus
 - side
 - failihooldus
 - lisautiliidid

6. Esimene tutvus

NB! Suur- ja väiketähtede koodid on U-s erinevad

date DATE Date

NB! Käsud ainult väiketähtedega!

Enter

P-i peatamiseks CTRL+C (Delete, Break)

viip shell valmis vastu võtma Sinu kāske

\$

username@ene:

\$ **NB! Ära ise hakka viipa sisestama! (Enter)**

kasutajanimi (username) ja parool (password)

administraator

Seanss:

login: username

Password:

You have mail

\$

\$ date

Thu Feb 6 09:58:46 GMT-2:00 1997

\$ logout

exit

CTRL+D lõpetab job-id, aga ka faili

There are stopped jobs sule ükshaaval

kui muidu ei saa, siis vaata protsessi # ja anna käsk

\$ kill #

\$ passwd parooli muutmine

6-8 alfanumeric

U-i käsud on keerulised

üldkuju

\$ käsk [võtmed] [argumendid]

abiinfo saamiseks *man*

\$ man command prints out the manual for a command

\$ man passwd

dokumentatsioon

\$ who

\$ who am i

Põhilised käsud:

cat - concatenates files (and just prints them out)

ex - text editor

finger - user information lookup program

ls - list contents of directory

mail - send and receive mail

passwd - change login password

who - who is on the system

write - write to another user

man -k mail - programs about mail

logout - logout

8. Failid

loogiliselt seotud nimega varustatud andmekogum

U-s faili mõiste laiem (kuvar, printer, ...)

3 tüüpi:

- harilik
- kataloog
- erifail (seadmed, harilikult /dev)

1. harilik - teksti või 2-nd

tekstiF-s vastab igale baidile 1 sümbol, sisaldab reavahetuse
märke

andmevahetuse korral U ja PC vahel tuleb reavahetusele
lisada kelgu tagasilükke märk

nimi, koht, teenistuslik info F kohta (omanik, pikkus, IDE, ...)
F nimi < 255 märki, üldjuhul laiendil erilist tähendust ei ole
ära alusta nime “.“-ga, väldi eritähendusega sümboleid (/ * ? - ‘ “ >)
mõned kompilaatorid tahavad kindlat laiendit (.p .pas)
nimi ei määra F tüüpi
mõnikord aitab (directory, C-s kirjutatud P)
\$ file failinimi
Reeglina mistahes P käivitamine - pöördumine vähemalt 1 F poole
failid K-des
U failiS
puukujuline, failipuu
K - temas sisalduvate F-de loetelu
jooksev K üldiselt uus F jooksvasse K
koduK, S-i sisenemisel on jooksev K koduK
/ 2 tähendust
/ - juur
absoluutne Fnimi, suhteline Fnimi
/export/home/user6/aavo/yles1.p yles1.p
Reegel: igal pool, kus kus võib kasutada harilikku F-nime, võib
kasutada ka absoluutset F-nime
liikumine käsuga cd - **c**hange **d**irectory
\$ cd kuhu
absoluutne, suhteline
\$ cd “koju“
\$ cd .. 1 nivoo / poole
\$ cd bin
\$ cd /bin erinevus?
\$ pwd **p**rint **w**orking **d**irectory
/export/home/user6/aavo
\$ cd ~aavo

\$ pwd

/export/home/user6/aavo

/bin põhiP-de K

/dev eriF-de K

/tmp ajutiste F-de K

K sisu kuvamiseks käsk

\$ ls list contents of directory

ls [-aAbcCdfFgiLmnopqrRstux1] [file...]

24 võtit

tähtsamad:

-l pika formaadiga list

-a ka varjatud F-d

-t sorditud aja järgi (modif. aeg)

-i väljastab F-ide IDE-d

-F F-nimele lisatakse F-i tüüpi iseloomustav märk (/ *)

\$ mv (**move**) F nime muutmine, ümbertõstmine

\$ mv ptk peatykk

NB! kui peatykk enne oli, siis ta kaob

\$ cp (**copy**) F kopeerimiseks

\$ cp ptk ptk1.1

Aga:

\$ cp ptk ptk1.1 raamat - K-i raamat

\$ rm (**remove**) F kustutamiseks

\$ rm peatykk ptk1.1

\$ cat (**concatenate**) F-de väljastamine, ühendamise

\$ cat file.3

\$ cat f1 f2 f3 F-d ühendatakse

\$ mkdir (**make directory**) luua K

\$ mkdir ained

\$ rmdir (**remove directory**) kustuta K (ainult tühja!)

9. Argumentide maskeerimine

argumentide (harilikult F-nimed) maskeerimine

* asendab mistahes sümbolite kombinatsiooni
abc* - abc, abc1, abc2, abca, abc3z4d
*.c, *.p

? asendab mistahes sümbolit
a? - aa, ab, a1, a9, a.,

[...] ab[xyz] - abx, aby, abz
pt[123] - pt1, pt2, pt3
par[1-37-9] - par1, par2, par3, par7, par8, par9

NB! P ei saa midagi aru *, ?, [...], seda teeb shell

\$ echo * väljastab jooksva K kõikide F-de nimed

\$ rm temp*

temp1, temp2,...

NB! \$ rm *

Shabloonid seatakse vastavusse ainult olemasolevate F-nimedega !

\$ mv ptk* peatykk* ei tööta, kuna pole olemas peatykk*

10. I/O ümberadresseerimine

I/O enamiku käskude puhul terminali kaudu

Reegel: terminali võib asendada I, O või mõlema operatsiooni jaoks failiga

O - >

\$ ls > faililist

\$ ls f1 f2 f3 > temp uus F

\$ ls f1 f2 f3 >> temp lisatakse 'temp' lõppu

\$ mail ain vaike toomas < teade_24

annab kvalitatiivselt uue võimaluse

\$ who > temp

\$ sort < temp tarbijate sorditud list

\$ who > temp

\$ wc -l < temp tarbijate arv (ridade arv)

\$ who > temp

\$ grep aavo < temp teeb kindlaks, kas tarbija 'aavo' on S-s

NB! > ja < töötleb shell, mitte vastav P

\$ sort < temp 1.

\$ sort temp 2.

1. sort ei võta 'temp'-i argumendina, ta sordib stand. I-voogu

2. sort saab 'temp'-i argumendina

11. Konveierid

1 P O \longrightarrow 2. P I

2 viisil:

- läbi (ajutise) F (temp)

- konveier

| - tähistus

\$ who | sort tarbijate sorditud listi väljastus

\$ who | wc -l tarbijate arvu väljastus

\$ ls | wc -l F-de väljastus

nii kuitahes palju P-e

P-e täidetakse konveieris üheaegselt, võimalik on dialoog

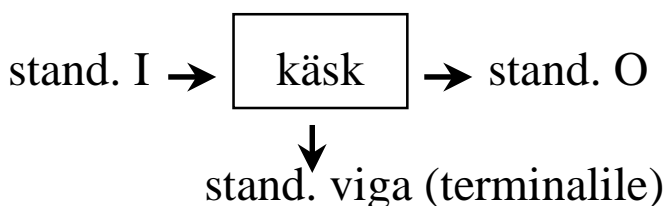
kui F-de nimesid ei ole näidatud, siis käsk sisestab stand. I-voogu

I-voog võib tulla:

- terminalilt

- failist

- konveierist



12. Shell

interpretaator

U - P-de ja andmete kogumik

mistahes käsk antakse interpretaatorile shell

- shell analüüsib käsku
- teeb vajalikud asendused
- otsib käsule vastava P-i
- annab S-le korralduse käsu täitmiseks

pärast käsu täitmist antakse juhtimine tagasi shell'ile

U toetab mitmeid interpretaatoreid, levinumad:

- Korn shell (AIX, ..) ksh
- Bourne shell (klassikaline) sh
- C shell (SCO U, SunOS, Solaris) csh, tsch

vaikimisi kasutatava määrab S-i administraator

tarbija saab startida selle shell'i, mida tahab, lõpetamine - exit

\$ echo \$SHELL

/bin/tcsh

U lubab keskkonda häälestada

U F-iS

fail on baitide jada

\$ ed

a

now is the time

for all good people

.

w junk

36

q

\$ ls -l junk

-rw-r--r-- 1 matti teachers 36 Feb 19 09:45 junk

\$ od -c junk

0000000 n o w i s t h e t i m e \n

0000020 f o r a l l g o o d p e o

0000040 p l e \n

0000044

od octal dump

-c baite sümbolitena

\$ od -cb junk

0000 n o w i s t h e t i m e \n
156 157 167 040 151 163 040 164 150 145 040 164 151 155 145 012

0020 f o r a l l g o o d p e o
146 157 162 040 141 154 154 040 147 157 157 144 040 160 145 157

0040 p l e \n
160 154 145 012

0000044

\n (012) - rea vahetus LF Enter

T kujutab 'Return'-i terminalil 2 sümboli summana

puudub F lõpu sümbol (T peab meeles F pikkuse)

P-d saavad andmeid F-st *read*'i abil (T P)

\$ file /bin /bin/ed

/bin: directory

/bin/ed: ELF 32-bit MSB executable SPARC Version 1...

file ei tee välja nimest - ta loeb mõnisada 1.-st baiti

```
$ od /bin/ed |more
```

```
0000000 077505 046106 000402 000400 000000 000000 000000 000000
0000020 000002 000002 000000 000001 000001 014714 000000 000064
0000040 000000 137170 000000 000000 000064 000040 000005 000050
```

...

077505 - P

tekstiF-de puhul otsitakse #include, mis viitab C-le

enamik käske töötleb faile sõltumatult nende tüübist (kui võimalik)
tegelikult faili tüüp puudub

universaalne - omal vastutusel (grep ei saa hakkama 2-F-dega,)

K-d ja F-de nimed

jooksev K

```
$ pwd
```

jooksev K on protsessi atribuut; kui pr-l on järglane, siis temal sama
kui aga järglane muudab jooksvat K, siis vanemat see enam ei
puuduta

jooksva K-i põhitähtsus U-s on organisatsiooniline

failipuu pilt

ls'ga saab kaevata, kuid saab ka otsida

graafilised puu esitusP-d

```
$ du disk usage (jooksva K-i K-de pikkused plokkides)
```

```
18  ./fm/export/home/user6/toomas
```

```
90  ./fm/export/home/user6
```

```
92  ./fm/export/home
```

```
94  ./fm/export
```

```
96  ./fm
```

```
$ du -a kõigi F-de pikkused
```

K-d on ka F-d

\$ head -8 fnimi	8 esimest rida
\$ tail -8 fnimi	8 viimast rida
\$ od -c fnimi	sümbolitena
\$ od -cb fnimi	sümbol-ja 8-nd
\$ od -cx fnimi	sümbol-ja 16-nd

F-de kasutusload

igal F-l 9 loabitti

3 kasutajakategooriat:

- u (user) omanik
- g (group) grupp
- o (other) teised

3 tüüpi load:

- r (read)
- w (write)
- x (execute) käivitada või otsida

vaatamiseks:

\$ ls -l fnimi

F omanik saab loabitte muuta

\$ chmod 'who' 'opcode' 'mode' [faililist]

who = ugo

opcode = + - =

mode = r w x

mode võib anda ka 8-arvuna

0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwX

\$ chmod go+w fnimi

\$ chmod 755 fnimi

\$ chmod 700 fnimi

\$ ls -ld . jooksva K olek

F omanikul ja super kasutajal (superuser) on õigus omistada

F-le uus omanik ja grupp

\$ chown uusomanik fnimi

\$ chgrp uusgrupp fnimi

F-i indeks

teenistuslik info (õigused, kpv.-d) koos süsteemse infoga
(pikkus, asukoht) nn. i-node'is

F i-node'i saab vaadata

\$ ls -i

788630 a.out	789498 mat.p	789495 sis2.dat
--------------	--------------	-----------------

789770 aaa	789507 matprot	789768 t1
------------	----------------	-----------

788506 bin	789521 matprot.doc	788816 t2
------------	--------------------	-----------

käskude cp, mv, ...puhul i-node muutub

\$ ls -l viimane muutmine

\$ ls -lu viimane kasutamine

\$ ls -lc viimane i-node muutmine

EriF-d

seadmete F-d

/devices

crw-----	1 root	sys	68, 11 Jan 14 14:01	eeeprom@1,f2000000:e...
----------	--------	-----	---------------------	-------------------------

crw-----	1 root	sys	74, 0 Jan 14 14:01	profile:profile
----------	--------	-----	--------------------	-----------------

68 - seadme tüüp

11 - jrk. nr.

SeosF-d

link - seos F-nime ja F vahel

1 F-ga võib mitu linki

tugev ja nõrk seos

tugev (hard) võimalik ainult 1 F-süsteemi piires

nõrk (soft, symbolic)

Failide tüübid:

- harilik(normal)
- d kataloog (directory)
- l link(link)
- s ühendus (socket)
- p konveieri (pipe)
- b plokk (block device)
- c sümbol (character device)

```
-rw-r--r-- 1 matti      2586 Nov  3 1995 parim.p
-rwxr-xr-x 1 matti     15044 Nov 14 1995 parimf
-rw-r--r-- 1 matti     2990 Nov 14 1995 parimf.p
-rw-r--r-- 1 matti      352 Nov  3 1995 prim.dat
drwxr-xr-x 2 matti       512 Apr 24 1996 public_html
-rw-r--r-- 2 uuno     74899 Dec 19 1995 unialg
-rw----- 1 matti     2955 Feb 14 1996 uuskaust
-rw-r--r-- 1 matti    16384 Jan 30 15:20 uvunix
```

harilikul F-l harilikult rw-r--r--
K-l harilikult drwxr-xr-x

```
      luba  sid. arv
omanik grupp teised  omanik  pikkus    viim. mod.      nimi
- rwx rwx rwx  1  matti 16384 Jan 30 15:20 uvunix
      421
      40
      20
      10
400   (r,-)
200   (w,-)
100   (x,s,-)
-,d,l,s,p,b,c
```

13. Protsessid

shell lubab

\$ date; who mõlemad käsud täidetakse

võib ka mitu protsessi samaaegselt

esiplaani (foreground) ja fooni (background) protsessid

\$ wc uunix > wc.uunix &

[1] 2735 protsessi # PID

\$

& - teade shell'ile

käivitatud P-i igat eksemplari nim. protsessiks

2735 - viimasele P-ile vastava protsessi # konveieri puhul

\$ kill 2735 fooniprotsessi 2735 lõpetamiseks

\$ kill 0 lõpetab kõik, v.a. lähteinterpretaatori protsessi

kill "tapab" ka kõik protsessi järglased

protsessidel hierarhiline struktuur (vanem, järglased)

mõned käsud:

\$ ps

PID	TTY	TIME	CMD
-----	-----	------	-----

3074	pts/4	0:02	tcsh
------	-------	------	------

\$ nohup *käsk* & (**no hangup**)

käsku täidetakse lõpuni (isegi pärast exit'it)

\$ nice *käsk* & alandab prioriteeti

\$ at 3am

käsk1

käsk2

...

Ctrl+D

\$

U-i T

Kuidas töötab U-i T?

töö peaorganisaator

- org-b protsesside täitmise järjekorda
- RAM'i reserveerimist
- kettaruumi kasutamist
- andmevahetust RAM'i ja perifeeriaseadmete vahel
- võtab protsessidelt vastu tellimusi teenindamiseks

T-ks nim. OS'i RAM'is paiknevat residentset osa

U'i T realiseerib suhteliselt piiratud f-de ringi

ta ei realiseeri midagi, mis vahetult teenindab tarbijat

tuum - ca 10000 rida C-s ja 1000 rida Assemblerit

lühike, 1 inimese jõud käib üle, kohandatav

enamik teisi lõplikul kujul, mittekohandatavad

T koosneb 2 põhisektsioonist:

- protsesside juhtimise (PJ)
- seadmete juhtimise (SJ)

PJ-s:

- reserveeritakse ressursid
- määratakse protsesside täitmise järjekord
- võetakse vastu teenindamise tellimused

SJ-s kontrollitakse andmevahetust RAM-i ja perifeeria vahel

* U-i üks plüsse: erinevatel arvutitel PJ sama

* SJ-s on 1 moodul iga perifeeriaseadme jaoks

* S-i genereerimine - kindlale perifeeriaseadmete hulgale
vastava U-i formeerimine

põhitöö genereerimisel - listi loomine, kus on täpsed
seadmete ja programsete vahendite kirjeldused

T sisaldab mitu tähtsat tabelit, mille abil koordineeritakse
omavahel seotud täidetavaid ahelaid

U - P, mille stuktuur on määratud andmetega

Suur osa tuuma tööst: tabelites otsimine ja nende modif.
mistahes hetkel A täidab:

- kas tarbija P-i
- või OS-i käsku

T-l protsesside tabel

S-i ühtsuse seisukohast tähtsaim on süsteemne kell (SK), mis
perioodiliselt formeerib katkestuse

katkestus - aparaatne signaal, mis lülitab arvuti täitma spetsiaalset
teenindusP-i; protsesside prioriteetide ümberhindamine
teiste katkestuste puudumisel SK realiseerib aja kvantimise f-i,
võimaldades A-t jagada hulga tarbijatega

multiP-ne ja swapping

protsessid konkureerivad omavahel CPU aja pärast
peatamine

protsess on objekt, mis asub T spets. tabelis (prots.-de tabel)

alglaadimine, protsessid 0 ja 1

T laadimine RAM-i ja tema käivitus - alglaadimine

protsess 0 - jõuga, fiktiivne

0 - andmestruktuur, mille abil T org-b multiP ja juhib protsesse
pärast seda S loob protsessi 1

alul 0-i koopia, "elustatakse", exec /etc/init

juhtimine U dispetscherile