Unix ja shell-ohjelmointi, luennot 1 & 2

1 Unix-käyttöjärjestelmä

1.1 Historia

Unixin ensimmäinen versio, nimeltään UNICS (*UNI*plexed operating and *C*omputing *S*ystem), luonnosteltiin ja toteutettiin AT&T:n Bellin laboratorioissa (Bell Laboratories) vuonna 1969. AT&T oli jo aiemmin (1964–1969) ollut mukana mm. General Electricin, Honeywellin ja Massachusetts Institute for Technologyn (MIT) kanssa MULTICS -projektissa, jonka yhtenä tarkoituksena oli selvittää yleiskäyttöisen, monen käyttäjän käyttöjärjestelmän toteuttamis- ja toimintamahdollisuuksia. Yksi suurimmista projektin rahoittajista oli USA:n puolustusministeriön tutkimustoimisto ARPA (Advanced Research Projects Agency) ja MULTICSin suunnittelussa kiinnitettiinkin runsaasti huomiota myös järjestelmän turvallisuuteen. MULTICS-projekti jäi kuitenkin pahasti jälkeen aikatulussa, eikä se ollut pystynyt täyttämään muutenkaan sille asetettuja, ehkä liiankin kovia, vaatimuksia. AT&T vetäytyi projektista 1969, perustellen sitä tavoitteellisilla erimielisyyksillä projektin osapuolten välillä. AT&T kertoi sen päätavoitteen olleen yksinkertaisesti luoda mahdollisimman käyttäjäystävällinen käyttöjärjestelmä.

UNICSin, josta siitemmin pian muotoutui UNIX, ensimmäiset kehittäjät olivat MULTICS-projektissakin mukana olleet Ken Thompson ja Dennis Ritchie. Ensimmäinen kone, jossa UNIX toimi oli DEC PDP-7.

Ensimmäistä kertaa UNIX toimi käytännön projektissa Bellin laboratorioiden patenttiosastolla tekstinkäsittelyjärjestemänä vuonna 1971. Kone oli DEC PDP-11, jossa muistia oli yhteensä 24 kilotavua: 16 kilotavua järjestelmälle ja 8 kilotavua käyttäjien ohjelmille. Levytilaa järjestelmässä oli 512 kilotavua ja suurin yksittäinen tiedosto oli rajattu 64 kilotavuun. Projekti oli onnistunut, ja se mahdollisti UNIXin jatkokehityksen ja laajemman käyttöönoton.

Vuonna 1973, C-kielen saadessa tietotyypit ja -rakenteet, UNIXin ydin (*kernel*) kirjoitettiin ensimmäisen kerran C-kielellä. Tämä oli siihen aikaan ennenkuulumatonta. Vaikka ydin olikin C-kielellä kirjoitettuna 20-40 % suurempi ja hitaampi kuin kokonaan konekielinen toteutus, oli juuri tällä valinnalla suuri merkitys UNIXin tulevaan suosion kasvuun: järjestelmästä tuli näin helposti ymmärrettävä, muutettava ja siirrettävä toisiin koneisiin.

Samana vuonna 1973 UNIX-asennusten määrä Bellin laboraroriolla nousi jo 25:een ja ensimmäinen tukiryhmä UNIX System Group perustettiin. Liittovaltion kanssa tehdyn sopimuksen mukaan AT&T ei vielä tuolloin saanut markkinoida tietokonetuotteita. Yhtiö tarjosi kuitenkin UNIX-järjestelmän sitä pyytäneille koulutuksellisiin tarkoituksiin.

Vuoteen 1977 mennessä asennusten lukumäärä oli noussut jo 500:een, joista 125 oli yliopistoilla. Mikroprosessoreiden suosion kasvu johti useat yhtiöt "porttaamaan" UNIX-järjestelmän uusiin koneisiin. Järjestelmän yksinkertaisuus ja helppo muokattavuus saivat aikaan useita UNIX-muunnoksia. Vuonna 1977 sai ensimmäisen versionsa myös Berkeleyn BSD (Berkeley Software Distribution), joka jatkossa saavutti vahvaa jalansijaa yliopisto- ja tutkijamaailmassa. Ensimmäinen BSD oli PDP-koneelle, mutta jo seuraavana vuonna julkaistiin versio VAX-koneelle.

Vuosina 1977-1982 Bellin laboratoriot yhdistelivät useita AT&T muunnoksia yhdeksi järjestelmäksi, joka sai nimekseen UNIX System III. Seuraavana vuonna AT&T julkaisi sen parannellun version UNIX System V. AT&T:n UNIX System V:stä tuli toinen Unix-järjestelmien päähaara edustaen Unixien kaupallista puolta, BSD:n edustaessa tutkimuspuolta. Vuoden 1984 alussa Unix-asennuksia arvioitiin yhteensä olevan 100000.

Unix-järjestelmä	**Myyjä/kehittäjä**	**Ympäristöt**
AIX	IBM	IBM RS6000, Motorola PowerPC
AU/X	Apple Computers	Motorola 68x
BSD	Berkeley Software Development	VAX, Intel x86
Digital Unix	Digital Equipment Corporation	Digital Alpha
FreeBSD	Berkeley Software Development	VAX, Intel x86
HP-UX	Hewlett Packard	HP PA RISC
IRIX	Silicon Graphics	MIPS
Linux	GNU Public License	Intel x86, Motorola 68k, Digital Alpha, Motorola PowerPC
Solaris	Sun Soft, Sun Microsystems	Sun UltraSPARC / SPARC, Intel x86, Motorola PowerPC

Taulukko 1: Muutamia Unix-versioita

Tänä päivänä Unix-asennuksia on maailmassa useita miljoonia. Tarkkaa lukua on lähes mahdoton arvioida johtuen mm. ilmaisten Intel x86 koneisiin sopivien Unix-kloonien (ennen kaikkea Linuxin[1]) suuresta suosion kasvusta 90-luvulla. Ei-kaupallisten kloonien lisäksi lähes kaikki suuret tietokoneyh-

[1] alunperin suomalaisen Linus Torvaldsin kehittämä Unix-klooni

tiöt ovat vuosien mittaan kehittäneet omat kaupalliset Unix-versionsa.

Eri Unix-versioiden yhteensopivuutta on yritetty säilyttää ja parantaa useiden standardien kuten ANSI C, IEEE POSIX, Open Software Foundation (OSF) jne. avulla. Tästä huolimatta jokainen järjestelmä sisältää omia erityispiirteitä niin rakenteen kuin systeemikutsujenkin osalta ja tämä vaikeuttaa ohjelmien siirrettävyyttä versioiden välillä.

1.2 Ominaisuuksia

- Unix on monen käyttäjän moniajokäyttöjärjestelmä. Käyttäjiä voi olla siis yhtäaikaa useampia, ja he voivat suorittaa useita, myös samoja ohjelmia samanaikaisesti.
- Käyttäjän kannalta Unix on työkalukeskeinen käyttöjärjestelmä. Runsaasta valikoimasta löytyy työkaluohjelmat lähes jokaiseen (pieneenkin) tehtävään. Tämä ja ohjelmoitavat *komentotulkit* (*shell*) mahdollistavat taas suurempienkin ohjelmien "kokoamisen" hyvinkin helposti.
- Unix on tyypillisesti laitteistoriippumaton ja se on kirjoitettu lähes kokonaan C-kielellä. Tämä mahdollistaa (koko) systeemin helpon muokattavuuden tarvittaessa.
- Unix sisältää vahvat tietoliikenneominaisuudet. Lähes kaikki versiot sisältävät suuren määrän tietoliikennettä hyödyntäviä ohjelmia, kuten sähköposti-, pääteyhteys- ja tiedostonsiirto-ohjelmia.
- Unix on vahvasti dokumentoitu käyttöjärjestelmä. Lähes kaikkiin ohjelmiin liittyy erilliset manuaali- ja / tai infosivut. Lisäksi saatavilla olevat lähdekoodit ovat (yleensä) hyvin dokumentoituja.

1.2.1 Käyttäjä

Koska Unixit ovat usean käyttäjän käyttöjärjestelmiä, täytyy jokainen käyttäjä pystyä tunnistamaan / yksilöimään. Tämä tapahtuu *käyttäjätunnuksen* (*account, username*) ja (useimmiten) *salasanan password* avulla. Käyttäjän *sisäänkirjoittautuminen* järjestelmään tapahtuu jonkin *päätelaitteen* ja / tai *pääteohjelman* avulla. Käyttäjän tunnistus voi tapahtua salasanan lisäksi myös esimerkiksi RSA-autentikointia (ssh) tai vaikkapa älykorttia tai sormenjälkitunnistusta käyttäen.

Jokaisella käyttäjällä on myös *käyttäjänumero* (*user ID, UID*), jota järjestelmä ja prosessit käyttävät käyttäjän yksilöimiseen. Lisäksi jokaisella käyttäjällä on *(primääriinen) ryhmänumero* (*group ID, GID*), *kotihakemisto*, (*home directory*) sekä komentotulkki, joka käyttäjän sisäänkirjautuessa suoritetaan.

Sisäänkirjautuessaan (login) Unix-järjestelmään käyttäjä aloittaa *istunnon* (*session*). Istunto lopetetaan kirjautumalla ulos järjestelmästä (logout). Kuten ohjelmia, niin istuntojakin voi olla käyttäjällä käynnissä useampia yhtäaikaa.

Unix-järjestelmät sisältävät suuren määrän apu- ja varusohjelmia. Vuosien varrella lähes kaikkia ajateltavissa olevia pikkutehtäviäkin varten on tehty oma ohjelma. Tämä voi olla käyttäjän näkökulmasta myös huono puoli – ohjelmien (jopa perustoimintoja suorittavien) omaksuminen ja opetteleminen vie aikaa. Toisaalta, ohjelmoitavien ja mukautuvien komentotulkkien ja apuohjelmien avulla voidaan helposti toteuttaa vaativiakin tehtäviä suorittavia ohjelmia: komentotiedostoja, shellskriptejä (*shell script*).

1.2.2 Merkkipohjainen käyttöliittymä

Unixin *käyttöliittymä* on perinteisesti ollut merkkipohjainen. Tämä tarkoittaa, että kaikki käyttäjän "näkemä" muodostuu ASCII-merkeistä. Tyypillisesti käyttäjän kirjautuessa sisään järjestelmään esimerkiksi verkon yli pääteohjelman avulla, suoritetaan käyttäjälle määritelty komentotulkki, jolloin käyttäjän ruudulle ilmestyy komentotulkin *kehote*. Tämän kehoitteen perään käyttäjä voi antaa komentonsa, esimerkiksi antaa suoritettavaksi haluttavan ohjelman nimi.

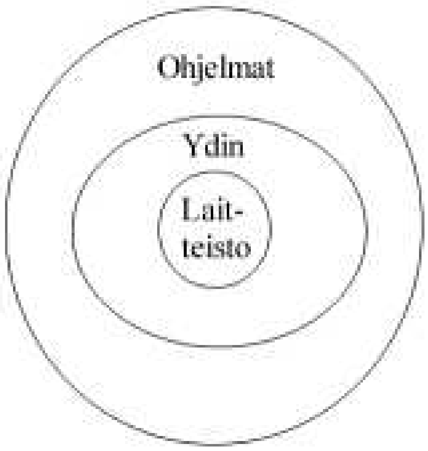
1.2.3 Graafinen käyttöliittymä

Merkkipohjaisen liittymän rinnalle on nykyisin tullut lähes kaikkiin Unix-versioihin saatavilla oleva *X Window System*. X Window System käyttää graafista ikkunointia, joka mahdollistaa grafiikkaa käyttävien / hyödyntävien ohjelmien käytön sekä esimerkiksi komentojen kätkemisen kuvakkeiden ja valikoiden taakse. X-ikkunointi perustuu palvelin-asiakas-malliin. *X asiakkaat* (*X client*) ovat joko lokaalisti tai etäkoneessa suoritettavia ohjelmia. X asiakas -ohjelmat esittävät syöte- (esim. näppäimen painalluksia, hiiren osoittimen paikka jne.) ja (graafisia) tulostuspyyntöjä lokaalilla työasemalla sijaitsevalle *X palvelimelle* (*X server*). X palvelimen tehtävänä on siis huolehtia näytön piirrosta sekä syötteiden välityksestä suoritettavalle ohjelmalle.

1.3 Rakenne

1.3.1 Ydin (*kernel*)

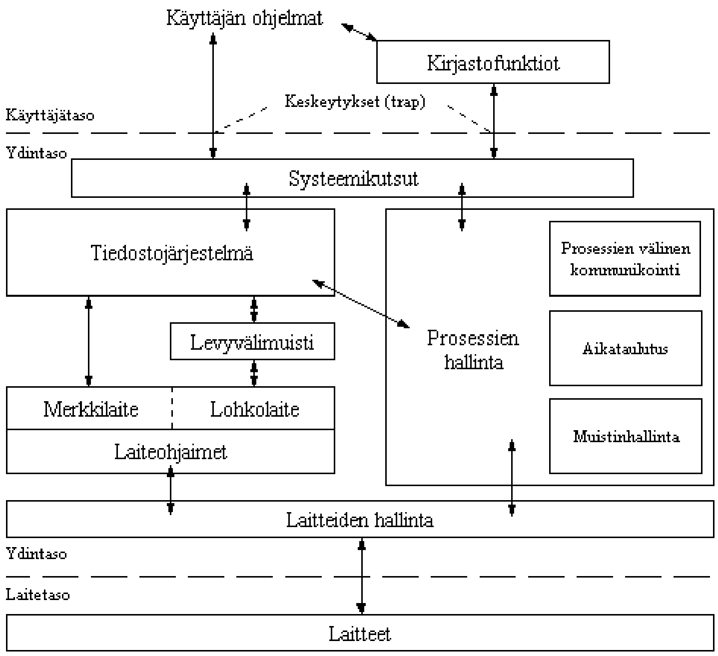
Unix-käyttöjärjestelmän sydän on *ydin* (*kernel*). Ydin on ohjelma, joka toimii rajapintana laitteiston ja sovellusten välillä. Kommunikointi sovellusten ja ytimen välillä tapahtuu *systeemikutsujen* avulla. Ytimen tehtävänä on on tarkistaa sille esitetty pyyntö (systeemikutsu), suorittaa sen vaatimat toimet ja mahdollisesti palauttaa jotakin pyynnön esittäjälle.



Kuva 1: Ydin toimii rajapintana ohjelmien ja laitteiden välillä.

Ytimen rakenne sisältää kaksi selvää alisysteemiä. Toinen alisysteemi huolehtii *tiedostojärjestelmästä* (sekä laitteista) ja toinen puolestaan *prosesseista* sekä niihin liittyvistä toiminteista. Prosessien hallinta -alisysteemi sisältää kolme tärkeää modulia. Yksi hoitaa prosessien välisen kommunikoinnin, toinen jakaa prosessoriaikaa prosesseille (aikatauluttaja) ja kolmas kontrolloi prosessien muistin varausta ja hallintaa.

Perinteisesti Unix-ytimet ovat olleet *monoliittisia*, ts. ydin on ollut yksi suuri ohjelma, joka on sisältänyt kaiken tarpeellisen (tuen kaikille laitteille jne.). Nykyään monoliittisten ratkaisujen rinnalle ovat tulleet ns. *mikroytimet*. Mikroytimet sisältävät ainoastaan käyttöjärjestelmän yleisten toimivuuden kannalta tarpeelliset toiminnot, jolloin ne ovat pienempiä ja toimivat nopeammin. Toisaalta eriytettäessä esimerkiksi laitetuki omiksi ohjelmikseen/moduuleikseen saadaan käyttöjärjestelmä mukautuvammaksi, eikä itse ydintä tarvitse muuttaa hankittaessa uusia laitteita. Tällä hetkellä suurin osa käytettävistä ytimistä on ns. *hybridejä*: ytimet ovat kohtuullisen kokoisia, mutta palvelu- ja laitetuki on pitkälle modulaarista.



Kuva 2: Ytimen rakenne

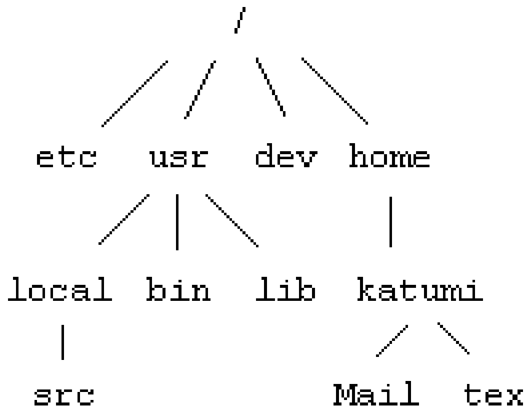
1.3.2 Tiedostojärjestelmä

Unixin tiedostojärjestelmälle on tyypillistä

- hierarkinen rakenne
- tiedostojen dynaamiset koon muutokset
- tiedon suojaus
- ulkoisten laitteiden käsittely tiedostoina.

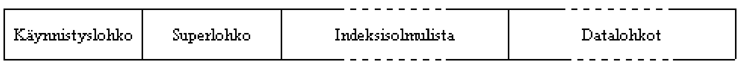
Unixin tiedostojärjestelmä on organisoitu puurakenteeksi, sisältäen yhden juurisolmun, jota merkitään '/' ja kutsutaan *juurihakemistoksi*. Juurihakemisto sisältää joukon Unix-järjestelmälle tyypillisiä hakemistoja, jotka puolestaan sisältävät tiedostoja ja alihakemistoja.

Unixin hakemistorakenne (*looginen* tiedostojärjestelmä) koostuu yleensä useammasta *fyysisestä* tiedostojärjestelmästä (*file system*). Fyysinen tiedostojärjestelmä liitetään johonkin hakemistopuun solmuun (alihakemistoon). Fyysinen tiedostojärjestelmä sijaisee *lohkolaitteella*, joka puolestaan on yksittäinen kiintolevy, sen yksittäinen *osio* (*partition, logical volume*), tai vaikkapa usean kovalevyn ryhmän (*volume group*) yksittäinen osio (*logical volume*). Eri Unix-versioilla on omat tiedostojärjestelmätyyppinsä ja ne ovat



Kuva 3: Esimerkki Unixin (loogisesta) tiedostojärjestelmästä

ominaisuuksiltaan usein hyvinkin erilaisia. Seuraavassa on esitetty Unixeille tyypillinen tiedostojärjestelmän perusrakenne.



Kuva 4: Fyysisen tiedostojärjestelmän rakenne

- Tiedostojärjestelmän ensimmäinen lohko on varattu käynnistyslohkoksi. Käynnistyslohko on käytössä ainoastaan levyillä/osioilla, joilta on tarkoitus pystyä käynnistämään koko Unix-järjestelmä.
- Toinen lohko ns. *superlohko* (*Super block*) sisältää tiedot levyn/osion rakenteesta, mm. indeksisolmujen ja datalohkojen määrän/koon.
- Kolmatta lohkoa kutsutaan indeksisolmu- tai indeksilistaksi (*inode list*), joka sisältää kaikki levyn/osion *indeksisolmut* (I-solmut, I-noodit). Indeksilistan koko määrätään *alustettaessa* tiedostojärjestelmä.
- Indeksilistan jälkeen tiedostojärjestelmälle on talletettuna kaikki tiedostojen sisältämä tieto, joka sijaitsee *datalohkoissa*. Datalohkon koko vaihtelee eri tiedostojärjestelmien välillä ja se voi olla kiinteä tai määrättävissä alustuksen yhteydessä. Datalohkon koko voi olla esim. 512, 1024, 2048 tai 4096 tavua.

Jokaisella tiedostojärjestelmän tiedostolla on oma indeksisolmunsa. Indeksisolmu sisältää tiedot mm. tiedoston *moodista*, *omistajasta*, *ryhmästä*, *koosta*, *muutosajoista* sekä kiinteästä määrä *osoitteita* tiedostolle kuuluviin datalohkoihin. Suurempien tiedostojen tapauksessa osoitteet voivat osoittaa datalohkoihin, jotka on varattu pelkästään "oikeiden" datalohkojen osoitteille.

owner:	katumi
group:	staff
type:	regular file
mode:	(0644/-rw-r–r–)
acessed:	Sun Jan 10 21:45:54
modified:	Sun Jan 10 21:45:47
changed:	Sun Jan 10 21:45:47
links:	1
size:	15044
disk addresses	

Kuva 5: Esimerkki indeksisolmusta

Usein sanotaan, että *"kaikki Unixissa ovat tiedostoja"*. Esimerkiksi haluttaessa ohjelmassa käyttää jotain laitetta, vaikkapa skanneria, ensin avataan laitetta *vastaava* tiedosto, ja sitten käytetään normaaleja tiedostonkäsittelyfunktioita. Unixin tiedostotyppeihin kuuluvat

- *Tavallinen tiedosto* (*regular file*). Esim. tekstitiedostot, erilaiset datatiedostot, suoritettavat tiedostot jne.

  ```
  -rw-r--r--  /home/k/katumi/unix/lecture1.tex
  -rwxr-xr-x  /bin/ls
  ```

- *Hakemisto* (*directory*).

  ```
  drwxr-x---  /root
  ```

- *Lohkolaite* (*block special*). Hajasaantinen, lohkottain luettava *laite*, joka voi sisältää tiedostojärjestelmän. Käyttää puskurimuistia (levyvälimuistia).

  ```
  brw-rw----  /dev/hda5
  ```

- *Merkkilaite* (*character special*), muut laitteet: nauhat, päätteet jne.

```
crw-------  /dev/isdnctrl0
```

- *Nimetty putki* (*named pipe, FIFO*). Prosessien välinen kommunikointikanava

```
prw-------  /dev/printer
```

- *Soketti* (*socket*). Prosessien välinen kommunikointikanava verkon yli eri koneiden välillä.

```
srwxrwxrwx  /tmp/.X11-unix/X0
```

- *Symbolinen linkki* (*symbolic link*). Tiedosto, joka sisältää toisen tiedoston koko nimen. Avattaessa avataankin linkkitiedoston sisällä lukeva tiedosto.

```
lrwxrwxrwx  /dev/ftape -> rft0
```

Lohkolaitteiden luku- ja kirjoitustoiminnot on *puskuroitu*. Ydin pitää viimeksi käytettyjä tietoja muistissaan ja voi palauttaa ne nopeasti (ilman levyltälukua), jos tieto löytyy vielä puskurista. Kirjoitustoiminnot puolestaan voidaan aluksi suorittaa nopeasti muistiin, josta ne puskurin täytyttyä tai sopivan ajan tullen kirjoitetaan fyysisesti levylle.

Useimmat Unix-tiedostojärjestelmätyypit sisältävät laitteita (laitetiedostoja) varten erilliset *pää-* (*major*) ja *alinumerot* (*minor*). Ydin sisältää taulukon laiteajureista ja päänumero on indeksi tähän taulukkoon. Yhdellä ajurilla voidaan hoitaa useampia laitteita, jolloin alinumeroa tarvitaan erittelemään mitä laitetta milloinkin tarkoitetaan. Alinumeroilla voidaan kuvata myös jonkin laitteen eri toimintoja. Esimerkiksi nauhurit sisältää usein omat alinumeronsa sen mukaan kelataanko nauha alkuun tiedoston sulkemisen jälkeen vai ei. Pää- ja alinumerot saa näkyviin esimerkiksi `ls -l tiednimi` komennolla.

```
brw-rw----   1 root     disk       3,   0 May  5  1998 /dev/hda
                                   ^    ^
                              päänumero alinumero
```

Tiedostojen *suojaus* tapahtuu Unixeissa ensisijaisesti *käyttäjä-* ja *ryhmä*tasoilla. Ainoastaan *ylikäyttäjällä* (*root*) on täydet oikeudet kaikkiin tiedostojärjestelmän tiedostoihin. Tiedostojen oikeudet, eli sen moodi voidaan määritellä erikseen sen omistajalle, ryhmälle sekä näihin kuulumattomille käyttäjille. Komennolla `ls -la tiednimi` saadaan näytölle seuraavanlainen tulostus.

X	oikeudet
0	ei oikeuksia
1	suoritus
2	kirjoitus
3	suoritus ja kirjoitus
4	luku
5	luku ja suoritus
6	luku ja kirjoitus
7	luku, kirjoitus ja suoritus

X	oikeudet
0	ei erikoisbittiä
1	sticky
2	sgid
3	sticky ja sgid
4	suid
5	suid ja sticky
6	suid ja sgid
7	suid, sgid ja sticky

Taulukko 2: Tiedoston moodi 0XXX (vasen) ja X000 (oikea)

```
-rw-r--r--   1 katumi     staff            0 Jan 13 19:51 foo
 ^^^^^^^^^     ^^^^^^     ^^^^^

 moodi         käyttäjä   ryhmä
```

Rivin alusta löytyy 10 merkin mittainen merkkijono. Kuten jo edellä kävi ilmi, ensimmäinen merkki ilmaisee tiedoston tyypin / laadun. Seuraavat 9 merkkiä ilmaisevat tiedoston *oikeudet*. Kolme ensimmäistä näistä ilmaisevat käyttäjän, kolme seuraavaa ryhmään kuuluvien ja kolme viimeistä muiden oikeudet. Tällaisen merkkikolmikon ensimmäinen merkki edustaa luku-, toinen kirjoitus- ja kolmas suorittamis- / selausoikeutta. Jos jokin merkeistä on '-', tarkoittaa se, ettei ko. oikeutta ole asetettu. Muussa tapauksessa merkki on `r`,`w` tai `x`.

Tiedoston moodi sisältää myös pari erikoistapausta (versiokohtaisia). Suoritettavalle tiedostolle voidaan asettaa ns. *suid* tai *sgid* -bitti (set user id, set group id). Suoritettaessa tällainen ohjelma toimii ko. tiedoston omistajan ja / tai ryhmän oikeuksilla. Tyypillisiä tapauksia ovat ohjelmat, joidenka jokin toiminto vaati esim. ylikäyttäjän (root) oikeuksia. Toinen erikoistapaus on ns. *sticky*-bitti, jonka asetus hakemistontapauksessa (useimmissa versioissa) estää käyttäjiä poistamasta tiedostoja, joihin heillä ei ole kirjoitusoikeutta.

Moodi ilmoitetaan ja asetetaan usein oktaalilukumuodossa. *Oikealta* kolme ensimmäistä lukua ovat järjestyksessä muut, ryhmä ja omistaja. Neljäs mahdollinen luku kertoo erikoistapaukset. Luvut täsmäävät oikeuksia seuraavasti.

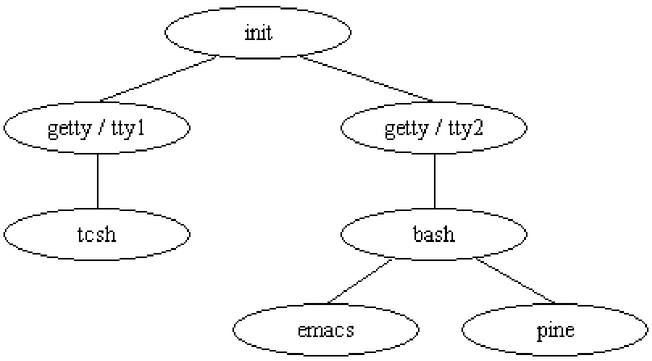
Esimerkiksi moodi 4755 vastaa oikeuksia:

```
-rwsr-xr-x   1 root     root            0 Jan 13 19:51 foo
```

Mainittakoon vielä, että joissakin Unix-tiedostojärjestelmäversioissa on myös käytettävissä ns. ACL-oikeudet (Access Control Lists), jotka mahdollistavat tiedostojen ja hakemistojen oikeuksien jakamisen juuri halutuille käyttäjille, ilman että käyttäjien tarvitsee olla samassa ryhmässä.

1.3.3 Prosessit

Tiedostojen lisäksi Unix sisältää vain yhden objektin: prosessin. Prosessit ovat ohjelmien suoritettavia *esiintymiä*. Moniajoympäristö ei rajoita ohjelmien yhtäaikaista suoritusta – samoja ohjelmia voidaan suorittaa useita samanaikaisesti. Unixissa jokaisella prosessilla on *vanhempi*prosessi. Ts. uusi prosessi syntyy, kun prosessi luo itselleen aliprosessin (lapsiprosessin). Unixin suoritettavista prosesseista syntyy siis *prosessipuu*. Prosessipuun juurisolmu on *init*-prosessi, joka luodaan kun systeemi käynnistetään. Kaikki prosessit on myös *odotettava* (*wait*). Jokaisen vanhempiprosessin tulisi "odottaa" kaikki luomansa lapsiprosessit. Jos lapsiprosessin suoritus loppuu ja vanhempi ei odota sitä, tulee prosessista *zombie*. Zombie-prosessi jää "roikkumaan" järjestelmään, kunnes vanhempi kuolee. Jos lapsiprosessin vanhempi "kuolee", tulee init-prosessista lapsiprosessin uusi vanhempi. Init-prosessi puolestaan odottaa koko ajan sille periytyneitä prosesseja.

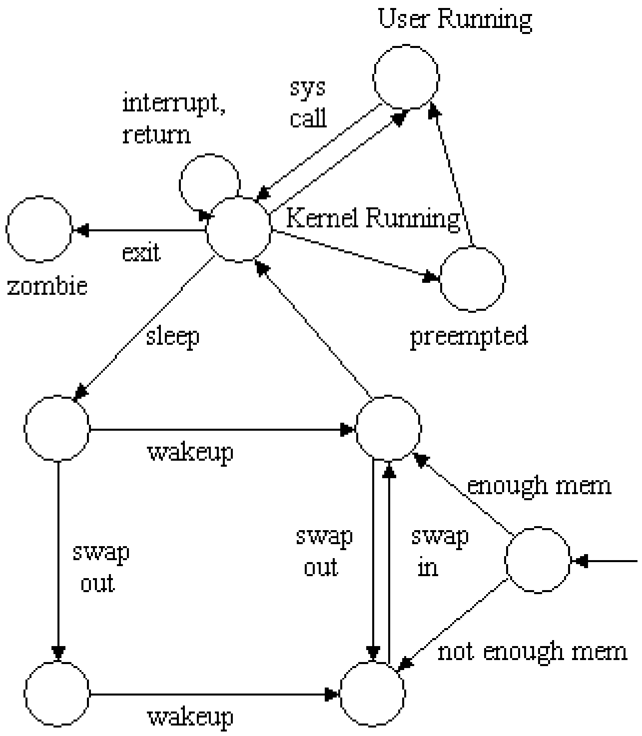


Kuva 6: Esimerkki prosessipuusta

Koska ydin huolehtii laitteistoresurssien jaon prosessien kesken, pitää se kirjaa kaikista olemassa olevista prosesseista *prosessitaulussa*. Taulu sisältää paljon tietoa jokaisesta prosessista:

- prosessin tila (mm. kerneltila, käyttäjätila, nukkuu)
- prosessin ID (PID) ja vanhempiprosessin ID (PPID)
- prosessin omistajan ID (PUID)

- efektiivinen käyttäjä (suid tai sgid)
- nykyhakemisto ja (juurihakemisto vrt. chroot)
- kontrollipääte
- *umask*, maski käyttäjän oletusmoodille (tiedostot)
- ajastimet ja hälytykset
- signaalimaskit (mitä signaaleita käsitellään)
- signaalivektori (osoittimet signaalinkäsittelijäfunktioihin)
- resurssien käyttö ja rajoitukset (limit)
- tiedostokahvat



Kuva 7: Prosessien tilat ja tilanvaihdot ytimessä

Prosesseihin liittyvät olennaisesti myös *signaalit*. Signaalien avulla prosessit voidaan saada reagoimaan (harvinaisiin) poikkeustilanteisiin ennalta suunnitellulla tavalla. Näin esimerkiksi *tapettaessa* prosessi voi vielä asettamansa *signaalin käsittelijän* avulla suorittaa toimenpiteitä, joiden suorittamatta jättäminen voisi johtaa ongelmatilanteisiin (esimerkiksi lukitukset ym.).

Prosessien välinen kommunikointi kuuluu Unixin vahvoihin puoliin. Tyypillisesti prosessit voivat kommunikoida (tavallisten tiedostojen lisäksi) mm. *putkien* (*pipe*, *named pipe*), *soketti*en (ja *domain soketti*en = koneiden välinen kommunikointi verkon yli) sekä *jaetun muistin* välityksellä.