

UNIX. Fitxategi-Sistema eta Komando-Interpretatzailea (KI)

Kontzeptuak

Izen absolutua eta erlatiboa, bidea (*path*), baimenak, jabea, loturak, dispositiboarekiko independentzia, berbideratzeak, pipeak, filtroak.

Enunziatua

UNIXeko fitxategi-sistema zuhaitz moduan dago antolaturik, bestelako loturak egin daitezkeelarik. Fitxategi sistema honen ezagutza eta maneia izango da lehen helburua: izenak, katalogoak, fitxategiak, nabigazioa sisteman zehar, loturak. Filtroen eta berbideratzeen bidez dispositiboarekiko independentzia ikusiko da. Komando-Interpretatzaileak eskaintzen duen hainbat aukera ere aztertuko dira.

Urratsak

Proposatutako ariketak egin, *bash* KI-rekin praktikatuz.

Dokumentazioa

- UNIXeko emandako apunteak.
- UNIXeko laguntza (*man*).

Linux-en oinarriak

Ezaugarriak

- + Sistema eragile multiprogramatua, erabiltzaile anitzekoa eta interaktiboa
- + Miniordenagailuetan eta lan-estazioetan estandarra
- + Komando-interpretatzaile ahaltsua
 - . berbideratzeak (< > >>)
 - . *pipe* komunikazioetarako (|)
 - . *background* edo atzeko mailako atazak (&)
- + Utilitate-programa asko
 - . aplikazio linguistikoak
 - . komunikazioak
 - . ezagutzaileak eta konpiladoreak sortzeko tresnak
 - . testu-prozesaketa
- Komandoak kriptikoak dira
- Informazio eta segurtasun gutxi sistematik

Komandoen sintaxia

Sinbolo bereziak:

- < berbideratu sarrera estandarra (adib: *prog < sarr_fitx*)
- > berbideratu irteera estandarra (adib: *prog > irt_fitx*)
- >> berbideratu irteera baina eransketa-moduan (adib: *prog >> era_fitx*)
- << sarrera datuak ondoren datoz (script-etan erabiltzen da)
- & atzeko mailako ataza, ondoren datozenekin konkurrentzian (adib: *prog &*)
- | ataza konkurrenteak: 1.aren irteera 2.aren sarrera (adib: *prog1 | prog2*)

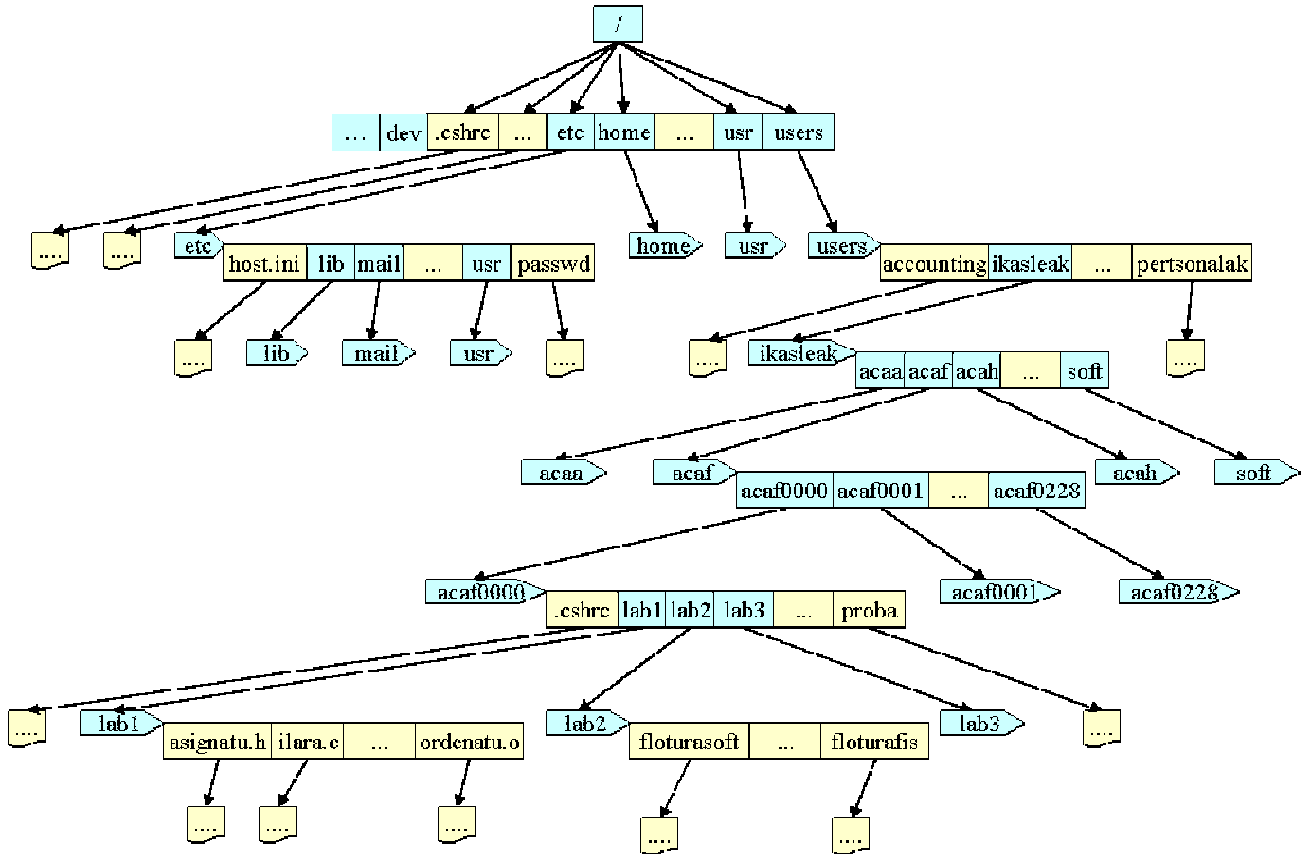
(Bakarrik beraien sarrera edo/eta irteera estandarra erabiltzen duten programen sarrera edo/eta irteera berbideratu daiteke. Era berean, pipe baten ezkerrean/eskuinean programa bat jartzeko bere irteera/sarrera estandarra erabili behar du)

Ordezkapen karaktereak (metakarakterek):

- * edozein karaktere-kate ordezkatzen du, baita kate hutsa ere
- ? karaktere bakarra ordezkatzen du

Bi karaktere hauetako bat komando bateko hitz baten agertzen bada, *shell*-ak (*bash* gure kasuan) hitza patroi bezala tratatuko du, patroia betetzen duten katalogoko sarrera guztien zerrenda alfabetikoaz ordezkatuz.

Fitxategi-sistema



- Zuhaitz erakoa, erro bakarrarekin (/).
- Unitateen muntaketa/dismuntaketaren beharra
- Fitxategien izena = [bidea/]sarrera
 - bide absolutua (erroarekiko) /kat1/kat2/kat3/fitx
 - bide erlatiboa (uneko katalogoarekiko) fitx
 - kat3/fitx
 - ../../../../kat2/kat3/fitx
 - ./ uneko katalogoa
 - ../ guraso katalogoa

Oinarrizko komandoak

- Shell-etik irteteko (lan saioa bukatzeko): *exit*, *logout*

exit

logout

- Gure erabiltzaile kontuaren pasahitza aldatzeko: *passwd*

passwd

- Komandoei buruzko laguntza: **man**

man komandoa

Komando edo funtzio bati buruzko laguntza ateratzen du irteera estandarrean.

Adibidea:

```
man ls
```

- Katalogoak listatzeko: **ls**

ls [aukerak] [kat]

Adierazitako katalogoaren sarrerak ateratzen ditu irteera estandarrean.

AUKERAK (batzuk bakarrik)

-a sarrera guztiak, puntuz (.) hasten direnak barne

-l informazio osoa: baimenak, loturak, jabea, taldea, tamaina, data, izena

-R azpikatalogoak ere listatzen ditu era errekursiboan

Adibideak:

```
ls
```

```
ls -a
```

```
ls -al
```

```
ls -l kat1
```

- Uneko katalogoa aldatzeko: **cd**

cd [katalogoa]

Uneko katalogoa adierazitakora aldatzen du. Helburu katalogoaren izena bide absolutua (errotik) edota erlatiboa (uneko katalogotik) bidez adierazi daiteke. Ez bada katalogorik adierazten, erabiltzailearen jatorrizko katalogora aldatzen du.

- Uneko katalogoa zein den jakiteko: **pwd**

pwd

Uneko katalogoaren bide absolutua erakusten du irteera estandarrean.

- Katalogo bat sortzeko: **mkdir**

```
mkdir [bidea/]katalogoa
```

Bidea ez bada adierazten, uneko katalogoan sortzen du katalogo berria.

- Katalogo bat ezabatzeko: **rmdir**

```
rmdir [bidea/]katalogoa
```

Katalogoa hutsik ez badago, ez du ezabatzen eta errore mezu bat erakusten du.

- Irteera estandarrean argumentuekin lotutako testu-mezuak ikusteko: **echo**

```
echo arg1 arg2 ... argN
```

- Fitxategiak kopiaatzeko: **cp**

```
cp fitx_iturri fitx_helburu
```

```
cp fitx_iturri1 ... fitx_iturriN kat_helburu
```

```
cp -R kat_iturri kat_helburu
```

Fitxategi bat kopiaatzeko balio du. Katalogo bat kopiaatzeko ere balio du. Baita fitxategi bat edo gehiago beste katalogo batean kopiaatzeko.

- Fitxategiei izena aldatzeko: **mv**

```
mv iturri helburu
```

```
mv iturri1 ... iturriN kat_helburu
```

Fitxategi edota katalogo bati izena aldatzeko balio du. Azken argumentua existitzen den katalogoa bada, orduan aurreko fitxategi guztiak duten izenarekin katalogo horretara mugitzen ditu.

- Fitxategiak ezabatzeko: **rm**

```
rm sarreral ... sarreraN
```

```
rm -R kat
```

```
rm -i fitx
```

Argumentu bezala pasatako fitxategi guztiak ezabatzen ditu. Katalogo ez hutsak ez ditu ezabatzen (-R adierazten bada, orduan errekurtsiboki katalogoa eta bere azpiko guztia ezabatzen du). -i adierazten denean baieztapena eskatzen du.

- Irteera estandarrean egutegia erakutsi: **cal**

```
cal [hila [urtea]]
```

Uneko hila erakusten du. Hila edo/eta urtea adierazten denean, eskatutakoa erakusten du.

- Fitxategien eduki osoa ikusteko (baita zenbait fitxategi kateatzeko ere): **cat**

```
cat fitx_zerrenda
```

Pasatako fitxategiak irteera estandarrean erakusten ditu, kateatuz. Ez bada argumenturik pasatzen, sarrera estandarra erakusten du.

Adibideak:

```
cat fitx1.txt
```

```
cat fitx1.txt fitx2.txt
```

- Fitxategien edukia pantailaz pantaila ikusteko: **more**

```
more fitx
```

- Fitxategien hasierako lerroak ikusteko: **head**

```
head [-n] fitx
```

Ez bada beste kopururik adierazten, lehen 10 lerroak erakusten ditu.

Adibideak:

```
head fitx1.txt
```

```
head -18 fitx1.txt
```

- Fitxategien bukaerako lerroak ikusteko: **tail**

```
tail [-n] fitx
```

Ez bada beste kopururik adierazten, azken 10 lerroak erakusten ditu.

Adibideak:

```
tail fitx1.txt
```

```
tail -8 fitx1.txt
```

- Filtroak: **grep**

```
grep [-v] patroia fitx_zerrenda
```

Pasatako fitxategien edukia aztertzen du, eta patroia duten lerroak erakusten ditu irteera estandarrean. Ez bada fitxategirik pasatzen, sarrera estandarra hartzen du sarrera bezala. -v adierazten bada, patroia ez duten lerroak erakusten ditu.

- Loturak: **ln**

```
ln [-s] exist_izena izen_berria
```

Hardware lotura: -s adierazi gabe. Dagokion katalogoan sarrera berri bat sortzen du, existitzen den sarreraren inodeari apuntatzen diona (**inodearen zenbakiaren bidezko lotura**).

```
ln exist_izena izen_berria
```

Software lotura: -s adierazita. Dagokion katalogoan sarrera berri bat sortzen du LINK motakoa, bere edukia existitzen den sarreraren izena delarik (**fitxategiaren izenaren bidezko lotura**).

```
ln -s exist_izena fitx_berria
```

- Fitxategiak editatzeko: **nano**, **emacs**, **vi**...

```
nano testu_fitx
```

```
emacs testu_fitx
```

```
vi testu_fitx
```

- Fitxategien eta katalogoen baimenak aldatzeko: **chmod**

```
chmod modua fitx_zerrenda
```

Pasatako fitxategiei irakurketa (r), idazketa (w), edota exekuzio (x) baimenak aldatzen die. Aldaketa jabearentzat (u), taldearentzat (g), beste erabiltzaileentzat (o) edota guztientzat (a) egin daiteke. Baimen hauek 9 biteko segida bezala kodeturik azaltzen dira fitxategi bakoitzeko.

- rwxrwxrwx

Jabea Taldea Besteak

Modua argumentuak baimen berriak adierazten ditu, bi eratara eman daitezkeelarik: hiru zenbaki ortzitar bezala (4: irakurketa, 2: idazketa, 1: exekuzioa), edota karaktere kate bezala (u|g|o|a nori adierazteko, +/- baimenak gehitu ala kendu egiten diren adierazteko, eta r|w|x zein baimen ematen edo kentzen diren esateko).

Adibideak:

```
chmod 740 *.txt      (rwx r-- ---)
```

```
chmod u+x fitx      (jabeari 'x' ipini)
```

Beste zenbait komando

- Konektatuta dauden erabiltzaileak ikusi: *who*, *w*
- Prozesu aktiboak ikusi: *ps*
- Prozesu bat amaitu: *kill*
`kill [-9] prozesu_zenbaki`
- Fitxategiak inprimatzeko prestatzeko, orritan banatzeko, identifikatzeko, zutabeak definitzeko, etab.: *pr*
- Fitxategiak inprimatzeko, normalean modu atzeratuan (*spooling*): *lpr*
`lpr -m - fitx1 (amaieran mezua)`
`lpr -Plaser fitx1 (laser izeneko inprimagailuan)`
- Inprimatze ilara ikusteko: *lpq*
`lpq -Plaser (laser izeneko inprimagailuan)`
- Baimen estandarrak: *umask*
- Fitxategiaren jabea eta taldea aldatzeko: *chown* eta *chgrp*
- Fitxategiak bilatzeko fitxategi-sisteman: *find*
`find / -name fitx -print`
- Irteera lehenetsia bikoizteko (irteera ikusi eta gorde nahi denean): *tee*
- Lerroak, hitzak eta karaktereak kontatu: *wc*
- Fitxategien lerroak alfabetikoki sailkatu: *sort*

- Unix mota: *uname*
- Martxan daraman denbora: *uptime*
- Fitxategi-sistemaren erabileraren egoera: *df*

Adibideak

<code>rm *.txt</code>	txt motako fitxategi guztiak ezabatzen ditu
<code>cat maiz?.txt</code>	izena "maiz"ez hasi, ".txt"ez bukatu eta tartean karaktere bakarra duten fitxategiak kateatu
<code>lpr maiz[1-3].txt</code>	maiz1, maiz2 eta maiz3 inprimatzen ditu
<code>cat fitx1 >fitx2</code>	<i>cp fitx1 fitx2</i> egitearen baliokidea
<code>head fitx1 >>buru</code>	fitx1 fitxategiaren lehenengo 10 lerroak metatzen ditu buru-ren bukaeran
<code>man cp >lis1 &</code>	cp-ren man orria kopiatzen du lis1 fitxategian, bitartean lanean jarrai daitekeelarik
<code>head -50 fitx1 lpr</code>	fitx1 fitxategiaren lehenengo 50 lerroak inprimatzen ditu

Proposatutako ariketak

1. Irten sistematik eta berriro sartu.
2. Aldatu zure pasahitza, eta egokiago bat ipini.
3. Exekutatu hurrengo komandoak:

```
man man
man echo
man 1 echo
```

4. Exekutatu hurrengo komandoak eta aztertu emaitza:

```
echo Kaixo zemuz zaude?
echo "Kaixo " zemuz zaude?
echo "Kaixo zemuz zaude?"
echo *
echo `*`
```

5. Uneko katalogoaren izena lortu, bide absolutua adieraziz.
6. Uneko katalogoaren edukia atera.
7. Uneko katalogoaren eduki *guztia* atera.
8. `/users/alumnos` katalogoaren edukia atera.
9. Zure kontuaren `home` katalogoan kokatu eta aztertu ea hurrengo komandoak berdinak diren edo ez:

```
cd ../../../../usr
cd /usr
```

10. Zure lan-katalogoan **Lab1** izeneko azpikatalogoa sortu.
11. Sortu berri duzun katalogoan kokatu (*Lab1* katalogoa) eta ondoren dagoen katalogo egitura sortu:

```
---|-- Programak ----|---- bin
|                       |---- datuak -----|---- urtarrila
|                       |                       |---- otsaila
|                       |                       |---- martxo
|                       |---- lib
|-- Gauzak
|-- Zaborra
```

12. *datuak* katalogoan kokatu. Aztertu zein katalogotara joaten zaren hurrengo komandoak exekutatzekoan (komandoak independenteak dira):

```
cd ../..
cd ../../Gauzak
cd ../Gauzak
cd urtarrila/../../Gauzak
cd urtarrila/../../Gauzak
cd /users/alumnos/acaf/acafxxxx/Lab1/Gauzak
cd $HOME/Lab1/Gauzak
cd
```

13. Zure kontuaren *home* katalogoan kokatu eta sortu software lotura bat (SOFT izenekoa) hurrengo katalogoari: `/users/alumnos/soft/acaf/eusk`

14. Software lotura bidez sortu duzun SOFT katalogora mugitu.

15. Berriro zure lan kontuko `Lab1` katalogoan kokatu eta `adibide.txt` izena duen fitxategia sortu (nano editorea erabili dezakezu). Fitxategiaren edukia, gutxienez, 40 lerrokoa izango da.

16. `adibide.txt` fitxategia `adibide1.txt` izena duen fitxategira kopiatu.

17. `adibide1.txt` fitxategia `Gauzak` azpikatalogoan kopiatu.

18. `adibide.txt` fitxategia `Gauzak` azpikatalogoan kopiatu `adibide2.txt` izenarekin.

19. `Gauzak` katalogoko `adibide2.txt` fitxategia `adibide4.txt` izenarekin berrizendatu.

20. `Lab1` katalogoan kokatu eta `Gauzak2` izeneko azpikatalogoa sortu.

21. `Gauzak` azpikatalogoko edukia `Gauzak2` azpikatalogoan kopiatu.

22. `Lab1` katalogoan kokatu, hurrengo komandoa exekutatu eta aztertu bere eragina:

```
cp -R Gauzak Gauzak3
```

23. Ondorengo komandoak exekutatu eta aztertu emaitzak:

```
cat adibide1.txt
more adibide1.txt
```

24. Makinara konektatuta dauden erabiltzaileen zerrenda lortu.

25. `Gauzak` katalogoan kokatu eta lortu `adibide1.txt` fitxategiko lehenengo 10 lerroak.

26. Lortu `adibide1.txt` fitxategiko azkeneko 10 lerroak.

27. Lortu `adibide1.txt` fitxategiko azkeneko 4 lerroak.

28. Lortu `adibide1.txt` fitxategiko 20 eta 30 arteko lerroak.

29. `adibide1.txt` fitxategiko lerro kopurua zenbatu.
30. `adibide1.txt` fitxategiko hitz kopurua zenbatu.
31. `adibide1.txt` fitxategiko karaktere kopurua zenbatu.
32. "kaixo" karaktere katea duten `adibide1.txt` fitxategiko lerroak lortu.
33. "kaixo" karaktere katea EZ duten `adibide1.txt` fitxategiko lerroak lortu.
34. `adibide1.txt` fitxategia `adibide3.txt` izenarekin kopiatu.
35. `adibide1.txt` fitxategia `adibide33.txt` izenarekin kopiatu.
36. `adibide1.txt` fitxategia `adibide333.txt` izenarekin kopiatu.
37. Ondorengo komandoak exekutatu eta aztertu emaitzak:

```
ls adibide*
echo ls adibide*
ls adibide?.txt
echo ls adibide?.txt
echo adibide?.txt
echo adibide*
```
38. `adibide333.txt` fitxategia ezabatu.
39. Ondorengo komandoa exekutatu eta aztertu emaitza:

```
echo rm *
```
40. Hilabete honen egutegia atera pantailan. Baita aurtengo urtekoa eta 1752. irailekoa ere.
41. Ondorengo komandoak exekutatu eta aztertu emaitzak:

```
cat adibide1.txt
cat adibide1.txt > emaitza.txt
ls
cat emaitza.txt
ls > emaitza.txt
cat emaitza.txt
```
42. `a1.txt` fitxategia sortu eta hiru testu lerro idatzi bertan.
43. Ondorengo komandoak exekutatu eta aztertu emaitzak:

```
cat a1.txt
cat a1.txt > emaitza.txt
cat emaitza.txt
cat a1.txt >> emaitza.txt
cat emaitza.txt
ls >> emaitza.txt
cat emaitza.txt
```