

***LABORATÓRIUMI ÚTMUTATÓ
AZ INFORMATIKA II. TÁRGYHOZ***

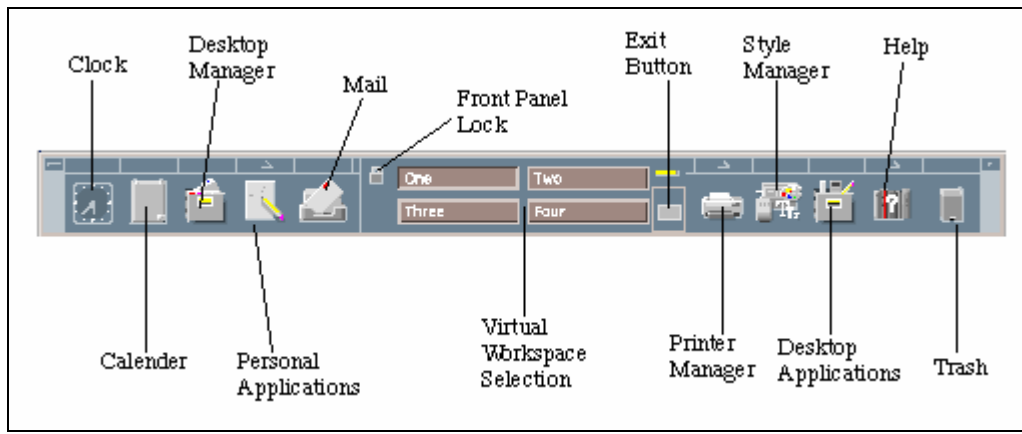
*készítette:
dr. Beinschróth József
2006.*

1. GRAFIKUS FELÜLETEK

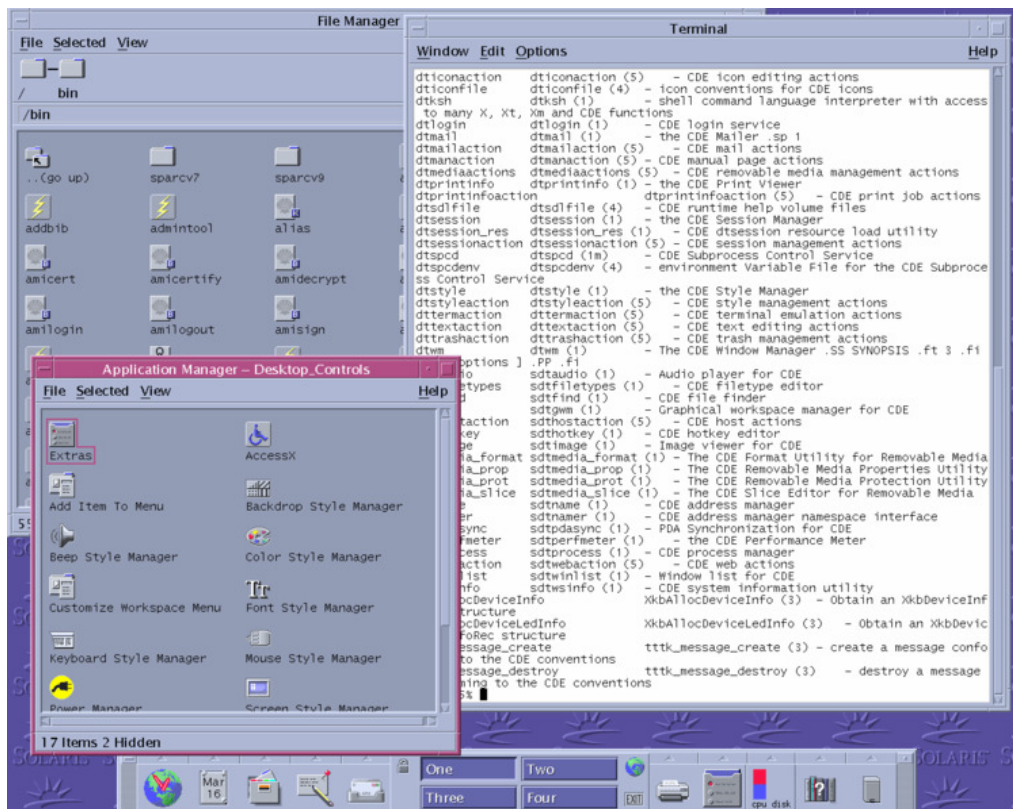
A Unix operációs rendszerek mindegyike rendelkezik grafikus felülettel, általában többel is. Ezek között igen elterjedt a CDE (Common Desktop Environment).

A bejelentkezéstől fogva a felhasználó a CDE desktopban dolgozik. A desktop része a front panel, a virtuális workspace-ek, és a window manager. A CDE támogatja az alkalmazások indítását a file managerből, az application managerből, és a front panelból.

A Front Panel:



A leggyakrabban használt alkalmazások:



FELADATOK:

1. Tanulmányozzuk a Front Panelen elérhető Helpet!
2. Tanulmányozzuk az egyes alkalmazások Helpjeit!
3. Szerezzünk gyakorlatot a grafikus felület használatában!
4. Tanulmányozzuk a következő alkalmazások használatát!

File Manager
Mail Tool
Terminal
Text Editor
Calendar
Calculator

2. EGYSZERŰ, ÁLTALÁNOS PARANCSONOK

who

Megadja a bejelentkezett felhasználók bejelentkezéssel kapcsolatos adatait: a userek bejelentkezési nevét, a kapcsolat azonosítóját, a bejelentkezés időpontját, remote login (távoli gépről történő bejelentkezések) esetén a távoli gép nevét is.

who am i

Megadja a parancsot kiadó user bejelentkezéssel kapcsolatos adatait, azokat az adatokat, amelyeket a who parancs kiadásakor az illető usernek megfelelő sorban kapnánk: a user bejelentkezési nevét, a kapcsolat azonosítóját, a bejelentkezés időpontját, remote login (távoli gépről történő bejelentkezés) esetén a távoli gép nevét is.

finger user[@host]

Részletes információt ad az argumentumban megadott userről, többek között megadja az illető user bejelentkezési nevét, home directory-ját, teljes nevét, utolsó bejelentkezésének időpontját és login shelljét. Akkor is működik, ha az illető user nincs bejelentkezve. Távoli gépen accounttal rendelkező user megadásakor a user nevéen kívül meg kell adnunk a távoli gép (remote host) nevét is, ekkor a user megadása user@host formában történik.

finger user3	a lokális gépen accounttal rendelkező user3 nevű userről ad információt
finger user3@arpad	az arpad gépen accounttal rendelkező user3 nevű userről ad információt
finger @arpad	információt ad az arpad gépre bejelentkezett userekről.

id [-a]

Megadja a parancsot kiadó usernek és elsődleges csoportjának nevét, valamint a user és a group azonosító számait (UID, GID).

- a nemcsak az elsődleges csoportot adja meg, hanem minden olyan csoportot, amelynek az illető user tagja

groups [user]

Megadja a parancsot kiadó user ill. az argumentumban szereplő user csoportjait.

write user

Üzenet küldés kezdeményezés egy ugyanazon gépre bejelentkezett user felé. A parancs kiadása után egy másodlagos prompt jelenik meg, ekkor van lehetőség az üzenet szövegének megadására. Az üzenet elküldése egy új sorban kiadott ^D-re történik meg.

wall

Körözvény küldés kezdeményezése az azonos gépre bejelentkezett valamennyi user felé. A parancs kiadása után egy másodlagos prompt jelenik meg, ekkor van lehetőség az üzenet szövegének megadására. Az üzenet elküldése egy új sorban kiadott ^D-re történik meg.

talk user[@host]

Interaktív kapcsolat kezdeményezés az argumentumban megadott user felé. A felhívott user jelzést kap a kezdeményezésről, értelemszerűen talk user-rel válaszol, ezután indul a kommunikáció. Távoli gépen regisztrált user felé történő kapcsolat kezdeményezésekor a user nevén kívül meg kell adnunk a távoli gép (remote host) nevét is, ekkor a userre való hivatkozás user@host formában történik. A kapcsolat felépülése után a userek akár egyszerre is begépelhetik üzeneteiket, mindkét monitoron láthatók lesznek a userek által beírt karaktersorozatok. A felépített kapcsolat bármelyik user által kiadott ^C-re lebomlik.

mesg [n/y]

A kapott üzenetek monitoron való megjelenítésének letiltása illetve engedélyezése, valamint - paraméter nélkül - a letiltott ill. engedélyezett állapot kijelzése. A letiltás nem vonatkozik a rendszeradminisztrátor üzeneteire.

mesg y	az üzenetek fogadására készen állunk, a továbbiakban az üzenetek megjelennek a monitoron (yes)
mesg n	az üzenetek fogadását letiltottuk, a továbbiakban az üzenetek - a rendszeradminisztrátor üzenetei kivételével - nem jelennek meg a monitoron (no)
mesg	megjelenítődik, hogy a továbbiakban az üzenetek megjelennek (y) vagy nem jelennek meg (n) a monitoron

passwd

Segítségével megváltoztathatjuk jelszavunkat. A rendszer először bekéri a régi jelszót, majd - az elgépelések kiküszöbölése érdekében - kétszer egymás után az újat. A jelszóként begépeltek karakterek nem jelennek meg a képernyőn. A userok csak saját jelszavukat változtathatják meg, a rendszeradminisztrátor azonban bármelyik userét -korábbi jelszavának ismerete nélkül is- módosíthatja.

man [-a][s fejezet] keyword

Kulcsszó szerinti keresés az Online Manual-ban. A rendszer a megadott keyword-öt (kulcsszót) egymás után keresi az Online Manual fejezeteiben. Ha talál az illető kulcsszóra vonatkozó információt, azt megadja és nem folytatja a keresést.

Az Online Manual az alábbi főbb fejezeteket tartalmazza:

- 1. fejezet: az átlagfelhasználó parancsai
- 1m fejezet: a rendszeradminisztrátor parancsai (maintanance)
- 2. fejezet: rendszer hívások
- 3. fejezet: C könyvtári hívások
- 4. fejezet: device leírások
- 5. fejezet: rendszer információs fájlok formátuma
- s keresés egy meghatározott fejezetben (section)
- a keresés valamennyi fejezetben (all)

man ls	az ls parancsra vonatkozó információ jelenítődik meg
man man	a man parancsra vonatkozó első információ jelenítődik meg
man -s1m mount	a mount parancsra vonatkozó, a rendszeradminisztrátornak szóló információ jelenítődik meg (az 1m. fejezetben levő információ)
man passwd	a passwd kulcsszóra vonatkozó információ jelenítődik meg
man -a passwd	a passwd kulcsszóra vonatkozó, bármelyik fejezetben előforduló információ jelenítődik meg
man -s5 passwd	a passwd kulcsszóra vonatkozó, a 5. fejezetben előforduló információ jelenítődik meg

cal [month/year]

Naptár megjelenítését teszi lehetővé. Paraméter nélkül az aktuális év aktuális hónapját, az év megadásakor az aktuális évet, az év és hónap együttes megadásakor a megadott év megadott hónapját jeleníti meg

cal 06	megjeleníti az i.sz. 6. évre vonatkozó naptárt
cal 2006	megjeleníti az 2006. évre vonatkozó naptárt
cal 02 2006	megjeleníti az 2006. februárjára vonatkozó naptárt

date [-u]

A rendszeridő és a dátum megjelenítése. A rendszeradminisztrátor ezt a parancsot megfelelően paraméterezve módosításokat is végrehajthat, a userek csak olvashatják a rendszeridőt és a dátumot

-u nem a lokális, hanem a greenwich-i idő megjelenítése

echo string

Az argumentumban szereplő stringet a képernyőre echozza. Használatának shell scriptek írásakor, vagy más parancsokkal összekapcsolva van értelme.

rup

Lista formájában megadja a lokális hálózati gépek futási és terhelési adatait.

rusers

Lista formájában megadja a lokális hálózati gépekre bejelentkezett userek bejelentkezési adatait a lokális gépre vonatkozó who parancshoz hasonló formában.

netstat -i

Információt ad a hálózati működésről.

last

Megadja az utolsó bejelentkezésekkel ill. a bootolással kapcsolatos információkat.

ping [host]

Távoli gép válaszának kérése.

/usr/sbin/ping arpad	válasz kérés az arpad géptől
/usr/sbin/ping 193.224.235.40	válasz kérés a megadott IP című géptől

spray [host]

Távoli gép válaszának kérése.

hostname

Megadja annak a host-nak a nevét, amelyre be vagyunk jelentkezve. A rendszeradminisztrátor ezen parancs segítségével meg is változtathatja a host nevét.

telnet [host]

Bejelentkezési lehetőséget biztosít a megadott távoli gépre. A bejelentkezés a távoli gépen accounttal rendelkező user nevének és jelszavának ismeretében lehetséges.

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Szomszédunkkal együttműködve (a telnet parancsot használva) próbáljuk ki a kommunikációs parancsokat (write, talk...)!
3. Mi a különbség a who am i és a whoami parancsok között?

4. Használjuk úgy a date parancsot, hogy az óra és perc értéken kívül mást ne jelenítsen meg!
5. Állapítsuk meg, hogy a hét melyik napján születtünk!
6. A man parancs felhasználásával olvassuk el a fenti parancsokhoz tartozó man page-eket!
7. Milyen szabályok betartásával választhatjuk meg jelszavainkat?
8. Mi az érdekessége az 1752-es év naptárának?
9. Értjük el, hogy egy távoli gépet saját gépünk folyamatosan „pingeljen”!
10. Állapítsuk meg, hogy milyen csoportokhoz tartozunk!
11. Tanulmányozzuk a bejelentkezett userek listáját!
12. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

3. FILE KEZELŐ PARANCSONK I.

cd [directory]

Az aktuális könyvtár beállítása

<code>cd /</code>	a root directory válik aktuális könyvtárrá
<code>cd /usr/bin</code>	a /usr/bin directory válik aktuális könyvtárrá
<code>cd ~</code>	a parancsot adó user home directory-ja válik aktuális könyvtárrá
<code>cd</code>	a parancsot adó user home directory-ja válik aktuális könyvtárrá, ugyanaz történik, mint a <code>cd ~</code> parancs hatására
<code>cd ~user</code>	a user nevű user home directory-ja válik aktuális könyvtárrá
<code>cd ~/user</code>	a parancsot adó user home directory-jában levő user nevű könyvtár válik aktuális könyvtárrá
<code>cd ..</code>	a jelenlegi aktuális könyvtár fölötti könyvtár válik aktuális könyvtárrá
<code>cd ../../</code>	a jelenlegi aktuális könyvtár fölött 2 szinttel magasabban levő könyvtár válik aktuális könyvtárrá
<code>cd ../dir1</code>	a jelenlegi aktuális könyvtár fölött levő könyvtárban található dir1 nevű könyvtár válik aktuális könyvtárrá
<code>cd .</code>	az aktuális könyvtár nem változik meg

pwd

Az aktuális könyvtár nevének, azaz az aktuális könyvtárhoz vezető útvonalnak a kiírása.

ls [-ladRFrtiu...] file1 file2 ...

Könyvtárak tartalmának ill. fájlok jellemzőinek különböző formátumok szerint történő listázása

- l hosszú (long) listát ad, a fájlok nevén kívül megjeleníti a típust, a hozzáférési jogosultságokat, a tulajdonost, a csoportot, a fájl méretét és az utolsó módosítás időpontját
- a nemcsak a normál, hanem a rejtett (ponttal kezdődő nevű) fájlokról is ad információt
- d az argumentumban megadott könyvtárról (és nem a tartalmáról) ad információt
- R rekurzív listázás, azaz egy alkönyvtárat is tartalmazó könyvtár listázásakor az alkönyvtár tartalma is kilistázódik

- F az egyes fájlokhoz tartozó sorokban a fájl típusára utaló jelölés is megjelenik, a * a végrehajtható, a / a könyvtári fájl jelöli
- r az enélküli esethez képest fordított sorrendű listázás
- t az utolsó módosítás időpontjának megfelelő sorrend szerinti listázás, a legutoljára módosított fájl jelenik meg a listában elsőként
- i az i-node-ok megjelenítése a listázás során az egyes fájlokhoz tartozó sorok első mezőjében
- u az utolsó módosítás időpontja helyett az utolsó hozzáférés időpontja jelenik meg az ls -l szerinti lista megfelelő mezőjében

ls	az aktuális könyvtárban levő fájl neveinek listázása
ls .	az aktuális könyvtárban levő fájl neveinek listázása
ls /	a root directory-ban levő fájl és könyvtárak neveinek listázása
ls -l ~	hosszú lista a home directory-ban levő fájlokról és könyvtárakról
ls -lR /	rekurzív hosszú lista a root directory-ból kiindulva
ls -lR / >file1	2>/dev/null rekurzív hosszú lista a root directory-ból kiindulva, a lista nem a képernyőre, hanem a file1-be kerül, a hibaüzenetek pedig eldobásra kerülnek
ls -lt /bin	az utolsó módosítás ideje szerinti sorrendű hosszú lista a /bin könyvtárról, a legutoljára módosított fájl lesz az első a listában
ls -lrt	az utolsó módosítás ideje szerinti hosszú lista az aktuális könyvtárról, a legkorábban módosított lesz az első a listában
ls -al /users	a /users könyvtár tartalmának a rejtett fájlkat is tartalmazó hosszú listája
ls -ld /users	a /users könyvtárra (nem a tartalmára) vonatkozó hosszú lista megjelenítése
ls -li ~user1 ~user2	a user1 és user2 userek home directory-jai által tartalmazott fájl és könyvtárak tartalmának hosszú listája a megfelelő i-node-okkal együtt
ls -lu	az aktuális könyvtár tartalmának listázása az ls -l formátumnak megfelelően, de nem az utolsó módosítás, hanem az utolsó hozzáférés időpontjának feltüntetésével
ls -lF ~/bin	a parancsot kiadó user home directory-jában levő bin alkönyvtárban levő fájl listázása, a végrehajtható típusúak *-gal lesznek megjelölve

cp [-ipr] file1 file2

Fájl másolás, a file1 másolataként létrejön a file2 fájl. A file2 tulajdonosa a másolást végző user, csoportja az illető user elsődleges csoportja lesz. A hozzáférési jogok rendszere is másolódik.

- i interaktív végrehajtás, ha a file2 már létezett, nem íródik felül automatikusan, a felülírás előtt visszakérdezés történik
- p nemcsak a fájl tartalma, hanem az utolsó módosítás időpontja is másolódik
- r rekurzív másolás, könyvtár másolása esetén a könyvtárak a bennük levő fájlokkal együtt másolódnak

cp local.profile ~/.profile a local.profile fájl a másolást végző user home directory-jába másolódik, ha ott a .profile már létezett, akkor visszakerdezés nélkül felülíródik

cp -i /etc/passwd ./passwd a /etc/passwd fájl az aktuális könyvtárba másolódik passwd néven, ha már volt ott ilyen nevű, akkor visszakerdezés történik

cp -rip /etc ~ a /etc könyvtár a másolást végző user home directory-jába másolódik, ott esetleg már létező fájlok felülírása előtt visszakerdezés történik

cp -rip /etc/profile /etc/env a /etc/profile és a /etc/env fájlok az aktuális könyvtárba másolódnak ott már létező fájlok felülírása előtt visszakerdezés történik

mv [-i] file1 file2

A file1 mozgatása file2-be, ugyanazon winchester partíción belül valójában átnevezés történik, a file1 át lesz nevezve file2-re. Ha az utolsóként megadott paraméter könyvtár, akkor a hivatkozott fájl megadott könyvtárba való átmozgatása történik. Ha mindkét paraméter könyvtár, akkor nem létező célkönyvtár esetén a forráskönyvtár átneveződik célkönyvtárrá, létező célkönyvtár esetén a forráskönyvtár tartalmával együtt átmozgatódik a célkönyvtár alá.

- i interaktív végrehajtás, ha a file2 már létezett, nem íródik felül automatikusan, a felülírás előtt visszakerdezés történik

mv textfile1 textfile2 a textfile1 neve textfile2-re változik, ha a textfile2 már létezett, akkor az eredeti textfile2 visszakerdezés nélkül felülíródik

mv /floppy/floppy0/f1 ~/pfile a floppy-n (a floppy root directory-jában) levő f1 nevű fájl a műveletet végző user home directory-jába mozgatódik és ott a pfile néven jön létre

mv -i textfile1 textfile2 a textfile1 neve textfile2-re változik, ha a textfile2 már létezett, akkor visszakerdezés történik

mv ~u1/f1 ~u1/f2 ~user2 az u1 nevű user home directory-jában található f1 és f2 fájlok átkerülnek a user2 home directory-jába

mv könyvtár1 könyvtár2 könyvtár1 mozgatása ill. átnevezése könyvtár2-be, ha könyvtár2 nem létezett, egyébként könyvtár1 mozgatása könyvtár2 alá

rm [-irf] file1 file2 ...

A paraméterlistában megadott fájlok törődnek, segítségével nem üres könyvtárak is törölhetők.

- i interaktív végrehajtás, a rendszer a törlés előtt megerősítést kér
- r rekurzív végrehajtás, ha a paraméterlistában nem üres könyvtár szerepel, akkor az tartalmával együtt törlődik
- f ha a műveletet végző user a megadott fájl tulajdonosa, de nincs rá írási joga, a kijelölt fájl akkor is törlődik

rm text	a text fájl törlődik
rm -i text	a text fájl törlődik, de törlés előtt a rendszer megerősítést kér
rm -r *	az aktuális könyvtár alatt levő fájlok és könyvtárak tartalmukkal együtt visszakérdezés nélkül vissza-állíthatatlanul törlődnek.

touch file

Az argumentumban megadott néve üres file jön létre.

mkdir [-p] könyvtár ...

Könyvtárak létrehozása, a paraméterlistában megadott könyvtárak létrejönnek.

- p ha a létrehozandó könyvtár olyan könyvtárban helyezkedik el, amely eddig nem létezett, akkor hiányzó szülő könyvtár, ill. könyvtárak automatikusan létrejönnek

mkdir ~/munka	a műveletet végző user home directory-jában létrejön egy munka nevű könyvtár
mkdir -p ~/k1/k2/k3/k4	a műveletet végző user home directory-jában létrejön a k1/k2/k3/k4 alkönyvtár struktúra

rmdir könyvtár ...

A paraméterlistában felsorolt üres könyvtárak törlődnek.

rmdir ~/munka1 ~/munka2	a műveletet végző user home directory-jában levő munka1 és munka2 könyvtárak -ha üresek- törlődnek.
-------------------------	---

ln [-s] file link

Segítségével szimbolikus vagy hard linket hozhatunk létre.

- s szimbolikus link jön létre

ln script1 script2	hard linket hozunk létre
ln -s ~/temp /tmp/temp	a műveletet végző user home directory-jában levő temp nevű könyvtárra mutató szimbolikus link jön létre a /tmp könyvtárban temp néven

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! A próbáknál alkalmazzunk joker karaktereket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Hozzunk létre file-okat, könyvtárakat és linkeket. Hol tehetjük ezt meg?
4. Másoljuk át a /usr/dict/words file-t a home direktorynkban létrehozott words nevű könyvtárba újwords néven. Hozzunk létre ugyanebben a direktoryban elhelyezkedő szimbolikus és hard linket!
5. Töröljük le a linkeket! Mi történik az eredetileg meglevő file-okkal?
6. Próbáljuk meg letörölni a /usr/dict/words file-t. Miért nem sikerül?
7. Mi a különbség az ls * és az echo * parancsok között?
8. Kérjünk listát a home direktory-nkban levő rejtett file-okról!
9. Milyen típusú file-ok vannak a /dev könyvtárban?
10. Hozzunk létre egy "proba1" fájlt! Készítsünk róla egy másolatot "proba2" néven!
11. A "proba1" fájlt mozgassuk el a /tmp könyvtárba!
12. Másoljuk át a /etc/default könyvtárat teljes tartalmával együtt a home direktorynkba!
13. Nézzük meg, hogy milyen fájlok vannak a home direktorynkban!
14. Ellenőrizzük le, hogy csakugyan a home direktorynkban vagyunk-e! Ha nem, akkor térjünk vissza oda!
15. Hozzunk létre egy proba-d nevű alkönyvtárat home direktorynkban !
16. Adjuk ki az ls -R / parancsot ! Mit tapasztalunk?
17. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

4. FILE KEZELŐ PARANCSONK II.

chown [-R] user file ...

Segítségével a paraméterlistában megadott fájlak tulajdonosa változtatható meg. Rendszeradminisztrátori parancs, a userek nem használhatják.

-R rekurzív, könyvtár esetén a tartalmazott fájlakra is vonatkozik

chgrp [-R] group file ...

Segítségével a paraméterlistában megadott fájlak csoportja megváltoztatható. A fájlak tulajdonosai fájljaikat hozzárendelhetik más csoportokhoz, de csak olyanokhoz, amelyekhez maguk is tartoznak. A rendszeradminisztrátor tetszőleges fájl tetszőleges csoportba áttehet.

-R rekurzív, könyvtár esetén a tartalmazott fájlakra is vonatkozik

chmod [-R] mode file ...

A paraméterlistában felsorolt fájlak hozzáférési jogosultságrendszerének megváltoztatása. A fájlak tulajdonosai fájljaik hozzáférési jogosultság rendszerét szabadon megváltoztathatják. A rendszeradminisztrátor tetszőleges fájl hozzáférési jogosultságrendszerét is szabadon megváltoztathatja. Az új jogosultságrendszer megadása történhet oktálisan és szimbolikusan.

Oktális megadás esetén a tulajdonos, a csoport és a többiek hozzáférési jogait rendre egy-egy oktális számmal adjuk meg, amely úgy alakul ki, hogy egy jog megléte esetén az azt reprezentáló jogot bináris egyesnek, meg nem létét bináris 0-nak feleltetjük meg és az rwx sorrend szerint így kialakuló háromjegyű bináris számot oktálisként fejezzük ki.

Szimbolikus megadás esetén meg kell adnunk, hogy mely tulajdonosi kategóriában (u: user, tulajdonos; g: group, csoport; o: others, a többiek; a: all, mindenki) az eddig meglévő jogokhoz képest milyen további jogokat adunk (+) vagy veszünk el (-), illetve, hogy a továbbiakban milyen jogok lesznek érvényesek (=).

-R rekurzív, könyvtár esetén a tartalmazott fájlakra is vonatkozik

chmod 777 file1	a file1-re minden tulajdonosi kategória minden joggal (rwx) rendelkezik
chmod 750 file2	a file2-re a tulajdonosnak minden joga, a csoportnak csak olvasási és végrehajtási joga, a többieknek pedig semmilyen joga nincs
chmod u+x file3	a file3-ra a tulajdonosnak az eddigi jogaihoz képest még végrehajtási joga is lett
chmod a=rwx file4	a file1-re minden tulajdonosi kategória minden joggal (rwx) rendelkezik

umask [mask]

Segítségével fájlok és könyvtárak default létrehozási maszkját kérdezhetjük le illetve állíthatjuk be. A maszk megadása háromjegyű oktális szám formájában történik. A megadott maszkból a könyvtárakra vonatkozó default hozzáférési jogosultság rendszer úgy alakul ki, hogy a megadott maszk oktális számjegyeit rendre 7-ből kivonjuk. A fájlokra vonatkozó default hozzáférési jogosultságrendszer hasonlóan számítható, azzal a különbséggel, hogy ha páratlan számot kapunk, akkor 1-et még kivonunk.

Az így kapott háromjegyű oktális számokat binárisként fogjuk fel, és a chmod parancsnál leírtak szerinti módon feleltetjük meg a fájlok és könyvtárak default hozzáférési jogosultság rendszerének.

Az umask és a fájlok valamint a könyvtárak default létrehozási maszkjai közötti összefüggéseket az alábbi táblázat tartalmazza.

umask	7-umask	könyvtár	fájl
0	7	rwx	rw-
1	6	rw-	rw-
2	5	r-x	r--
3	4	r--	r--
4	3	-wx	-w-
5	2	-w-	-w-
6	1	--x	---
7	0	---	---

Az így kialakuló hozzáférési jogosultságrendszerek egy része nem vezet praktikusan felhasználható kombinációra. A gyakorlatban is felhasználható hozzáférési jogosultság rendszerre a 0, 2 ill. a 7-es umask értékek vezetnek, ezért a többit nem szokás használni.

umask	megadja a maszk értékét
umask 027	a továbbiakban a könyvtárak 750, fájlok 640 oktális számokkal jellemzett hozzáférési jogokkal jönnek létre

setfacl [-msd] ACL-entry file...

ACL (Access Control List) jogok beállítása, módosítása, törlése.

- m ACL létrehozása
- s Meglevő jogosultság rendszer helyettesítése új ACL-lel
- d ACL törlése

ACL-entry	Jelentés
u::jogosultság	A tulajdonos jogosultságának beállítása, a jogosultságot oktálisan kell megadni
g::jogosultság	A csoport jogosultságának beállítása, a jogosultságot oktálisan kell megadni
o::jogosultság	A többiek jogosultságának beállítása, a jogosultságot oktálisan kell megadni
m::jogosultság	A maszk értékének beállítása, a jogosultságot oktálisan kell megadni
u:user_név:jogosultság	A user_névvel megadott user jogosultságának beállítása, a jogosultságot oktálisan kell megadni
g:group_név:jogosultság	A group_névvel megadott csoport jogosultságának beállítása, a jogosultságot oktálisan kell megadni

getfacl file...

ACL (Access Control List) jogok lekérdezése. (Az ls -l prancs nem jeleníti meg közvetlenül az ACL jogosultságokat, csupán egy + jellel jelzi létezésüket.)

find starting_pathname find_pattern command

Fájlok keresése a megadott könyvtár alatt rekurzíven, a megadott keresési minta szerint, a megadott parancs a talált fájlkon végre lesz hajtva. A starting_pathname (kezdő könyvtár) megadja azt a könyvtárat, ahonnan a keresés kezdődik. A find_pattern (keresési minta) segítségével a keresésre vonatkozó feltételeket adhatunk meg a következők szerint:

- | | |
|---|----------------------|
| -name file név | (pl. -name 'fil*') |
| -perm hozzáférési_jogok | (pl. -perm 644) |
| -type file típus: fdcb | (pl. -type f) |
| -user felhasználói_név | (pl. -user oktató) |
| -group csoport_név | (pl. -group oktatás) |
| -mtime módosítás n napon belül vagy kívül | (pl. -mtime 1) |

A minták negálhatók (!), egymással és (a) ill. vagy (o) kapcsolatba hozhatók, így meglehetősen összetett keresési feltételek alakíthatók ki.

Az argumentumban megadott command -parancs- a megtalált fájlokra vonatkozik és a következők szerint fog végrehajtódni:

- print file nevek kilistázása az ls parancsnak megfelelő formátum szerint
- ls file nevek kilistázása az ls -l parancsnak megfelelő formátum szerint
- exec megadott parancs végrehajtása a megtalált fájlokra
(pl. -exec rm {} \;)

```
find / -name .profile -ls az ls -l parancsnak megfelelő formátum szerint kilistázza
                           a .profile nevű fájlokat, a root directory-től kezdődően
                           végignézve az egész könyvtárrendszert
find /users -user root -printaz ls parancsnak megfelelő formátum szerint a
                           /users könyvtártól kezdődően kilistázza azokat a fájlokat,
                           amelyeknek a tulajdonosa a root
find /users \( -user user1 -a -mtime -1 \) -exec rm {} \;
                           törli azokat a fájlokat, amelyeknek a tulajdonosa a user1,
                           és 1 napon belül lettek utoljára módosítva
```

df [-k] [file-system ...] [file ...]

Diszk-partíciók teljes, felhasznált és üres területének kiírása blokkban ill. KB-ban. Nem teljes diszk partíció, könyvtár vagy fájl megadása esetén az illető könyvtárat vagy fájlt tartalmazó diszk partícióra vonatkozó adatok jelenítődnek meg.

-k táblázatos formában jelenít meg és az adatokat kByte-ban adja meg

```
df az összes diszk partícióra vonatkozó adatot megjeleníti
df / a root partícióra vonatkozó adatokat jeleníti meg
df -k /usr táblázatos információt ad a /usr-ről
```

du [-ksa] [file ...]

Fájlok, könyvtárak méretének meghatározása

- k a méreteket kByte-ban íratjuk ki
- s a könyvtárakra csak a rekurzív végösszeget íratjuk ki
- a nemcsak az alkönyvtárakra, hanem az egyes normál fájlokra is ad információt

mount

A könyvtárrendszerbe beépült, felmountolt fájl-rendszerek kiírása. A userek csak paraméter nélkül használhatják, így számukra csak információkérésre szolgál. A rendszeradminisztrátor megfelelő paraméterezéssel file-systemek felmountolására is alkalmazhatja.

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Állítsuk be az umask értéket 027-re. Milyen jogokkal jön létre ezután egy könyvtár és egy file? Meddig lesz ez a beállítás érvényes?
4. Home hozzunk létre egy könyvtárat és egy file-t! Állítsuk be úgy a hozzáférési jogokat, hogy bármelyik user képes legyen törölni ezeket! Kérjük meg a szomszédunkat törlésükre!
5. Határozzuk meg, hogy az egyes userek mekkora diszk területet használnak!
6. Vannak-e a root tulajdonában mindenki számára rwx jogosultsággal rendelkező file-ok?
7. Tanulmányozzuk a setfacl ill. a getfacl parancsokhoz tartozó man page-eket!
8. Hozzunk létre egy file-t. Állítsuk be, hogy senkinek ne lehessen több joga rá, mint r. (ACL)!
9. Hozzunk létre egy file-t. Állítsa be, hogy szomszédjának (de csak neki) rwx joga legyen rá!
10. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

5. FILTER PARANCSONK

Reguláris kifejezések (kiértékelésük jobbról balra történik)

grep style

.	egy tetsz. karakter
^	sor eleje
\$	sor vége
[...]	egy kar. a felsoroltak közül
[-]	intervallum megadása az ASCII kódtáblából (pl. a-z)
[^]	kivéve a ^ utáni karaktert
*	bármennyi az előtte állóból
\	spec jelentés visszavonása
\<	szókezdet
\>	szóvég

egrep style

*	0 vagy több az előtte álló reg kif-ből
?	0 vagy 1
+	1 vagy több
	vagy kapcsolat
()	csoportosítás

cat [file ...]

Fájlok összefűzését és megjelenítését teszi lehetővé, a bemenetére, a standard inputra kerülő fájlokat sorban egymás után kiteszi a kimenetére, a standard outputra.

cat file1	a file1 fájl tartalmát a képernyőre listázza
cat file1 file2	a file1 és a file2 tartalmát egymás után a képernyőre listázza
cat file1 > file2	a file1 tartalma file2-be listázódik, azaz a file1 átmásolódik a file2-be
cat file1 >> file2	a file1 tartalma file2 eddigi tartalmához hozzáfűződik
cat file1 file2 > file3	a file1 és a file2 egymás után bekerül a file3-ba
cat > file1	a billentyűzetről a file1 fájlba történő közvetlen beírást tesz lehetővé
cat - file1 >file2	a billentyűzetről a file2-be történő beírás után a file2-höz hozzáíródik a file1 tartalma
cat file1 - > file2	a file2-be beíródik a file1, majd hozzáíródnak a billentyűzetről bevitt karakterek

cat < file1	a file1 fájl tartalmát a képernyőre listázza, ugyanaz történik, mint a cat file1 esetén
cat	a billentyűzetről érkező karaktereket a képernyőre echozza
cat < /dev/term/a > /dev/term/b	az a soros portról beolvasott karakterek a b soros portra kerülnek
cat /etc/passwd > /dev/bpp0	a /etc/passwd fájl tartalma kinyomtatásra kerül

more [file ...]

Text fájlok laponként történő megjelenítését teszi lehetővé. Korlátozott, de jól használható önálló (belső) parancskészlettel rendelkezik, amelyekkel a megjelenítési és keresési funkciók válnak elérhetővé. A keresési funkciókban jól használhatók a reguláris kifejezések.

Az elérhető fontosabb parancsok a következők:

h:	segítség kérés, help
SPACE:	egy oldalt előre lép
RETURN:	egy sort előre lép
b, ^B:	egy oldalt vissza lép
q,Q:	kilépés
/pattern:	megadott reguláris kifejezés keresése

more /usr/dict/words a /usr/dict/words file tartalmát listázza a képernyőre, a képernyő vagy ablak teleírása után megáll, ekkor válnak elérhetővé az előzőekben leírt parancsok.

/sun kilistázza azokat a sorokat, amelyek a sun stringet tartalmazzák
/^sun kilistázza azokat a sorokat, amelyek a sun stringgel kezdődnek
/^yes\$ kilistázza azokat a sorokat, amelyekben pontosan a yes string található
/100\$ kilistázza azokat a sorokat, amelyekben a 100\$ string szerepel

grep [-v] regular_expression file ...

Lehetővé teszi reguláris kifejezések fájlokban történő keresését. Ha a parancs argumentumában szereplő reguláris kifejezés nemcsak karakterkonstansokat tartalmaz, hanem univerzális jelentéssel rendelkező karaktereket (*,., \$, stb) is akkor ezeket a shell elöl el kell rejtenünk. Ez megvalósítható például úgy, hogy a reguláris kifejezést aposztrófok közé tesszük. Gyakran nem önmagában, hanem csövekhez kapcsolva szokás használni.

-v	azokat a sorokat adja meg, amelyekben nem szerepel megadott string
-i	case insensitive
-v	azokat a sorokat adja meg, amiben nincs meg a minta
-n	a megtalált sor sorszámát adja meg

- l nemcsak a sorokat, hanem a file neveket is megadja, akkor vab értelme, ha az argumentumban több file is szerepel

grep sun /usr/dict/words	kilistázza /usr/dict/words fájlt azon sorait, amelyek a sun stringet tartalmazzák
grep '^sun' /usr/dict/words	kilistázza /usr/dict/words azon sorait, amelyek a sun stringgel kezdődnek
grep '^yes\$' /usr/dict/words	kilistázza /usr/dict/words azon sorait, amelyekben pontosan a yes string található
who grep root	megadja, hogy a root be van-e jelentkezve
ls -lR / 2>/dev/null grep '\.ps\$'	az egész könyvtárrendszert végignézve megkeresi a postscript (.ps-re végződő) fájlokat
ls -lRF /users/oktató 2>/dev/null grep '*\$'	a /users/oktató könyvtártól kezdődően rekurzív módon megkeresi a végrehajtható fájlokat
grep -v arpad /etc/hosts	azokat a sorokat adja meg, amelyekben nem szerepel az arpad string
grep -l arpad /etc/host /etc/passwd	megadja az argumentumban szereplő két file azon sorait, amelyekben szerepel az arpad string
grep -i ARPAD /etc/hosts	megadja azokat a sorokat, amelyekben szerepel az arpad string, akár kis, akár nagybetűkkel
grep ARPAD /etc/hosts	azokat a sorokat adja meg, amelyekben nagybetűvel szerepel az ARPAD string
grep '^girl' /usr/dict/words	a girl stringgel kezdődő sorokat listázza ki a /usr/dict/words-ből
grep 'lady\$' /usr/dict/words	a lady stringre végződő sorokat listázza ki a /usr/dict/words-ből

fgrep string file

Működése hasonló a grep parancs működéséhez, de ez nem értelmezi a reguláris kifejezéseket, stringként keres, ezáltal gyorsabb lesz a keresés.

fgrep sun /usr/dict/words	kilistázza /usr/dict/words fájlt azon sorait, amelyek a sun stringet tartalmazzák
---------------------------	---

egrep string file

Extended grep. Működése hasonló a grep parancs működéséhez, de ez jóval bonyolultabb reguláris kifejezéseket (egrep style) ill. logikai kapcsolatokat is elfogad.

head [-n] [file ...]

Megjeleníti az argumentumban megadott fájl első n sorát. Ha nem adjuk meg a megjeleníteni kívánt sorok számát, akkor az első 10 sort jeleníti meg.

head -5 /usr/dict/words	megjeleníti a /usr/dict/words fájl első 5 sorát
ls -l /etc head	megjeleníti a /etc könyvtárra vonatkozó, az ls -l parancs formátumának megfelelő lista első 10 sorát

tail [-n,/+m] [file ...]

Megjeleníti az argumentumban szereplő fájl utolsó n sorát, vagy az m-edik sorától az utolsó soráig terjedő tartományt. Ha nem adjuk meg sem a megjeleníteni kívánt sorok számát, sem a megjelenítendő tartomány kezdő sorát, akkor az utolsó 10 sor jelenik meg.

tail -5 /usr/dict/words	megjeleníti a /usr/dict/words fájl utolsó 5 sorát
ls -l /etc tail +20	megjeleníti a /etc könyvtárra vonatkozó, az ls -l parancs formátumának megfelelő listát a 20-ik sortól kezdődően
(head -1 /usr/dict/words; tail -1 /usr/dict/words) > ff az ff fájlba beírja a /usr/dict/words első és utolsó sorát	
head /usr/dict/words>ff; tail /usr/dict/words) >> ff az ff fájlba beírja a /usr/dict/words első és utolsó 10 sorát	

cut -cfd file

Kilistázza, „kivágja” egy file sorainak bizonyos karaktereit ill. mezőit.

-c number a megadott karakterpozíció(k)ban levő karaktereket adja meg
 -f number a megadott mező(k)ben levő karaktereket adja meg (szep:tab.)
 -d szeparátorjel megadása

cut -c1-6 /etc/passwd	kilistázza a /etc/passwd sorainak első 6 karakterét
cut -c1-3,5-7 /etc/hosts	kilistázza a /etc/hosts sorainak első 3 ill. 5-7 katakterét
cut -d. -f1-3 /etc/hosts	kilistázza a /etc/hosts sorainak első 3 mezőjét a . karaktert tekintve szeparátor jelnek

tr string1 string2

A string1-ben megadott karaktereket a string2 megfelelő karaktereire cseréli.

tr u U	Minden kis u-t nagy U-ra cserél a begépelt szövegben
--------	--

tr u U < /etc/hosts	A /etc/hosts file-ban minden kis u-t nagy U-ra cserél
tr 'abc' 'ABC'	A kisa-t nagy A-ra, a kis b-t nagy B-re a kis C-t nagy C-re cseréli, a többi katraktert változatlanul hagyja a begépelt szövegben
tr '[A-Z]' '[a-z]'	A nagybetűket a nekik megfelelő kisbetűkre cseréli a begépelt szövegben

wc [-cwl] [file ...]

Megadja az argumentumban szereplő fájl karaktereinek, szavainak és sorainak számát. Gyakran nem önmagában, hanem csövekhez kapcsolva szokás használni.

-c (character)	a fájl karaktereinek számát adja meg
-w (word)	a fájl szavainak számát adja meg
-l (line)	a fájl sorainak számát adja meg

wc /usr/dict/words	megadja a /usr/dict/words karaktereinek, szavainak és sorainak számát
wc -w /usr/dict/words	megadja a /usr/dict/words szavainak számát
who grep root wc -l	megadja, hogy a root hányszor van bejelentkezve
ls -lR / grep "\.au\$"	megadja, hogy az egész könyvtárrendszerben hány audio (.au-ra végződő) fájl van

sort [-nrfu] [file ...]

Elvégzi a megadott fájl sorainak a megadott szempontok szerint történő rendezését. Default esetben ASCII sorrendbe rendez a sorok elejétől kezdődően. Gyakran nem önmagában, hanem csövekhez kapcsolva szokás használni.

-n	számsorrend szerinti rendezés
-r	rendezés fordított sorrendben
-f	a rendezés folyamán nem lesz különbség a kis és a nagybetű között
-u	azonos sorokból csak egyet tart meg

sort /usr/dict/words	a /usr/dict/words fájl tartalmát abc sorrendben (az ASCII kódtáblázat szerint) a képernyőre listázza
sort -r /usr/dict/words >f1	a /usr/dict/words fájl tartalmát fordított abc sorrendben beírja az f1 fájlba
who sort	a bejelentkezett felhasználók bejelentkezési adatait a nevek abc szerinti sorrendben megjeleníti
du -sk /export/home/users/* sort -rn head -3	megadja a három legtöbb diszk területet használó user diszk használatra vonatkozó adatait

compress [file ...]

Az argumentumban szereplő fájlok tömörítését végzi a Lempel-Ziv kódolási algoritmus szerint. Az eredeti fájlt törli, a tömörített állományt file.Z néven hozza létre.

-d tetszőlegesen kis méretű fájlokat is tömörít

`compress /usr/dict/words` a `/usr/dict/words` fájlt tömöríti, tömörítés után az eredeti fájlt törli, a tömörített állomány `words.Z` néven jön létre

uncompress [file ...]

A `compress` paranccsal tömörített (.Z-re végződő) fájlok visszaállítását végzi. A fájl nevének megadásakor nem szükséges a .Z kiterjesztést megadni.

`uncompress /usr/dict/words` a `/usr/dict/words.Z` fájlt tömörített formából visszaállítja, visszaállítás után a tömörített állományt törli

awk

Összetett és funkciógazdag file feldolgozó program. (Lásd man `awk` !)

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! A próbáknál alkalmazzunk a reguláris kifejezéseket, a csöveket és az átirányításokat! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Hozzunk létre egy file-t (`cat` parancs), amelyben néhány nagyváros neve szerepel!
4. Szerepel-e a `/usr/dict/words`-ben az `apple` string?
5. Hány audio (.au végződésű) file-t tartalmaz a rendszer?
6. Keressünk `setuid` és `sticky` bitekkel ellátott file-okat!

7. Keressünk ACL-lel rendelkező file-okat!
8. Hány olyan file található a teljes file-systemben, ahol a tulajdonos hozzáférési jognak helyen nem x, hanem s szerepel?
9. Írassuk ki a pontos időt (óra:perc)!
10. Határozzuk meg saját user id-nket!
11. Írassuk ki egy sorba a bejelentkezett userek nevét
12. Írassuk ki egy sorba a futó processzek nevét!
13. Mit eredményez a következő parancs: `grep * file`?
14. Mit eredményez a következő parancs: `grep '^[a-zA-Z]' /usr/dict/words`?
15. Tanulmányozzuk az `awk` parancshoz tartozó man page-eket!
16. Határozzuk meg, hogy hány user van bejelentkezve!
17. Jelenítsük meg a következő fájlokat és tanulmányozzuk tartalmukat!
 - a. `/etc/passwd`
 - b. `/etc/group`
 - c. `/etc/hosts`
18. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

6. ARCHIVÁLÁSSAL ÉS NYOMTATÁSSAL KAPCSOLATOS PARANCSONOK

tar [-ctxvf...] opciók

Általánosan használt archiváló parancs, segítségével tetszőleges eszközre, winchesterre, fájlba, floppy-ra archiválhatunk, abszolút és relatív pathname-mel egyaránt. Mindig rekurzív módon működik. Az archiválásra kijelölt állományokat egyetlen fájlba fűzi össze, de nem tömörít.

- c tar fájl létrehozása
- t tar file listázása
- x tar file visszaállítása
- v archiválás közben folyamatos információ megjelenítése az archiválás előrehaladásáról
- f tar file megadása, hiányában a TAPE környezeti változó értéke lesz figyelembe véve

```
tar -c .az aktuális könyvtár archiválása a TAPE környezeti változó értékének megfelelő állományba, relatív pathname-mel
tar -cvf /floppy/floppy0/proba.tar ~
    A home directory archiválása, abszolút pathname-mel floppy-ra, az archivált állomány proba.tar néven jön létre, archiválás közben információt kapunk az archiválás folyamatáról
tar -tvf /floppy/floppy0/proba.tar a proba.tar fájlban archivált fájlok neveinek listázása az ls -l parancs formátumának megfelelően
tar -xvf /dev/rmt/0 a tape-en levő archivált állomány visszaállítása, abszolút pathname-mel történt archiválás esetén az eredeti helyre, relatív pathname-mel történt archiválás esetén az aktuális könyvtárba
tar -tvf /dev/rmt/0 a tape-en levő archivált állományban archivált fájlok neveinek listázása az ls -l parancs formátumának megfelelően, a parancs újabb kiadásakor a következő állomány listázása fog megtörténni
```

mt [-f tape_name] command

Segítségével parancsokat küldhetünk a tape drive-nak. A fontosabb parancsok a következők:

- | | |
|--------|--|
| eom: | a drive az utolsó archivált anyag végére áll |
| rewind | a drive a szalag kezdetére áll |
| status | információ megjelenítés a tape drive-ról |
| erase | a szalagon levő információ törlődik |

ufsdump

Rendszeradminisztrátori parancs, lehetőséget ad inkrementális file-system archiválásra. (Lásd man ufsdump!)

ufsrestore

Rendszeradminisztrátori parancs, lehetőséget ad inkrementális file-system visszaállításra. (Lásd man ufsrestore!)

lp [-d destination] [-o option] file1 ...

Az argumentumban felsorolt fájlokra vonatkozó nyomtatási kérelem előállítása. A fájlok az LPDEST környezeti változó értékének megfelelő printeren lesznek kinyomtatva

- d destination printer megadása, ebben az esetben nem az LPDEST környezeti változónak megfelelő printeren, hanem az argumentumban megadott printeren történik a nyomtatás
- o segítségével printer típustól függő opciók adhatók meg (nobanner, nofilebreak, length, width ...)

```
lp /etc/passwd      a /etc/passwd fájlra vonatkozó nyomtatási kérelem létrehozása.
                    A nyomtatás a default printeren történik
lp -d epson /etc/passwd a /etc/passwd fájlra vonatkozó nyomtatási kérelem
                    létrehozása. A nyomtatás az epson nevű printeren történik
lp -o nobanner /etc/env a /etc/env -re vonatkozó nyomtatási kérelem létrehozása.
                    A nyomtatás a default printeren történik
```

lpstat [-a] [printername1 ...]

A printerek és print jobok állapotának lekérdezésére szolgáló parancs

- a megadja, hogy az argumentumban megadott printerek, vagy ezek hiánya esetén a default printer elfogad-e nyomtatási kérelmeket

cancel request-id

A megadott print job törlése a default (LPDEST környezeti változóval megadott vagy az lp nevű) printer queue-ból

Printelés átirányítás felhasználásával:

cat file > /dev/bpp0

A file a párhuzamos portra kerül, ezáltal a nyomtatóra kerül.

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Archiváljuk (tar parancs) relatív pathname-mel home direktory-nkat kazettára!
A visszaállítást a /tmp könyvtárba végezzük!
4. Archiváljuk (tar parancs) abszolút pathname-mel home direktory-nkat szalagra. Töröljünk mindent a home direktory-nkból, majd állítsuk vissza az archivált anyagból.
5. Készítsünk mentést a /etc könyvtárról! A mentés kerüljön a home direktory-nkba!
6. Mi az oka a megjelenő hibaüzeneteknek?
7. Kérjünk listát a mentés tartalmáról!
8. Próbáljuk ki az mt parancsokat!
9. Tanulmányozzuk az ufsdumphoz ill. az ufsrestore-hoz tartozó man page-eket!
10. Megfelelő parancsokkal nyomtassuk ki a /etc/profile file-t!
11. Vizsgáljuk meg a nyomtatási sor tartalmát!
12. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

7. A VI EDITOR ÉS A SED EDITOR

7.1 vi editor

vi file

Üzemmodok

Beszúrási üzemmód
(Insert mode)

Parancs üzemmód
(Command mode)

Utolsó sor üzemmód
(Last line mode)

Üzemmodok közötti váltás

Beszúrási üzemmód
Utolsó sor üzemmód
Parancs üzemmód
Parancs üzemmód

-->Parancs üzemmód
-->Parancs üzemmód
-->Beszúrási üzemmód
-->Utolsó sor üzemmód

Esc
Esc
a,A,i,l,o,O
:

Kurzor pozícionálás (parancs módban)

h, balra nyíl	Egy karaktert balra
j, le nyíl	Egy sort le
k, fel nyíl	Egy sort fel
l, jobb nyíl	Egy karaktert jobbra
w	Egy szót jobbra
^F	Egy oldalt előre
^B	Egy oldalt vissza
G	Szöveg végére
1G	Szöveg elejére
10G	Tizedik sorba

Beszúrási módok

a	Beszúrás a kurzor mögé
A	Beszúrás a sor vége mögé
i	Beszúrás a kurzor elé
I	Beszúrás a sor eleje elé
o	Új sor beszúrása az aktuális sor mögé
O	Új sor beszúrása az aktuális sor elé

Szöveg törlése

x	A kurzor alatti karakter törlése
X	A kurzor előtti karakter törlése
dw	Az aktuális szó törlése (csak a kurzortól jobbra)
dd	Az aktuális sor törlése
D	Törlés a kurzortól a sor végéig
dG	Törlés a kurzortól a fájl végéig
d1G	Törlés a kurzortól a fájl elejéig
:5,10d	Törlés az 5.sortól a 10.sorig

Szöveg módosítása

cw	Az aktuális szó módosítása (csak a kurzortól jobbra)
cc	Az aktuális sor módosítás
C	Módosítás a kurzortól a sor végéig
J	Az aktuális sor egyesítése a következő sorral

u	Vissza az előző parancsot (Undo)
U	Vissza az e sorra kiadott összes parancsot
:u	Vissza az előző utolsó sori parancsot

Copy, Cut & Paste

dd	Az aktuális sor törlése a vágólapra (cut)
dw	Az aktuális szó törlése a vágólapra (cut)
yy	Az aktuális sor másolása a vágólapra (copy)
yw	Az aktuális szó másolása a vágólapra (copy)
p	Beillesztés az aktuális sor alá (Paste)
P	Beillesztés az aktuális sor fölé (Paste)
:1,10 co 20	1-10 sor másolása a 20. sor után
:1.10 m 20	1-10 sor mozgatása a 20. sor után

Find & Replace

/string	A megadott string keresése előre
?string	A megadott string keresése visszafelé
n	Az előző keresés ismétlése azonos irányban
N	Az előző keresés ismétlése ellenkező irányban
:g/search/s//replace/	A következő search string cseréje replace stringre
:g/search/s//replace/g	Az összes search string cseréje a replace stringre
g/search/s//replace/gc	A következő search string cseréje a replace stringre (global, consult)

Mentés és kilépés

:w	Mentés (Save)
:w newfile	Mentés más (newfile) néven (Save As...)
n:wq vagy :x vagy ZZ	Mentés és kilépés (Save & Exit)
:q	Kilépés (Quit)
:q!	Erőszakos kilépés (mentés nélkül)
:e newfile	Új fájl (newfile) megnyitása (Open)
:r newfile	Új fájl (newfile) beillesztése a kurzor mögé (Insert)

7.2 sed editor

Stream editor – nem interaktív editor, azaz editálsa közben nem jelenik meg a file a képernyőn. Tipikusan shell scriptekben szokás használni.

sed -enf 'transzformáció' file

- e edit
- n normal output elnyomás
- f file script

transzformáció: tartomány művelet

tartomány: cím
cím1, cím2

művelet: d: delete
 s: substitute
 r: read
 w: write
 p: print

példák műveletre: 2d 2,12d 17,\$d
/alma/d /[A-Z]\$, 100d/alma/,/korte/d

sed 's/east/west/' /data.file	A /data.file-ban az east stringet west stringre cseréli
sed 's/3/9/' /data.file	A /data.file-ban a 3-asokat 9-esekre cseréli (minden sorban az első előfordulásig)
sed 's/3/9/g' /data.file	A /data.file-ban a 3-asokat 9-esekre cseréli (mindet)

<code>sed ' /west/d' /data.file</code>	A /data.file-ban a west stringet tartalmazó sorokat törli
<code>sed '2d' /data.file</code>	A /data.file-ban törli a 2. sort
<code>sed '/^[]*\$ /d' /data.file</code>	A /data.file-ban törli az üres sorokat

A parancsok file-ba (sedscrip) írhatók, a file-ra a sed argumentumában hivatkozhatunk, így ugyanaz történik, mintha a sed argumentumába írtuk volna a parancsokat.

script: sedscrip

```
s/De.*son/Frank Marcus/
/west/d
s/[0-9] / /
```

```
sed -f sedscrip /data.file
```

FELADATOK:

1. Tanulmányozzuk a fenti parancsok man page-eit!
2. Hozzunk létre egy használható szöveget tartalmazó file-t!
pl. `man ksh > ksh.file`
3. Próbáljuk ki benne a különböző vi szerkesztési funkciókat!
4. Értjük el, hogy a vi editor kijelezze üzemmódjait!
5. Milyen gépek nevei szerepelnek a /etc/hosts file-ban?
6. Keressük meg a /etc/shadow file-ban a jelszó nélküli felhasználókat!
7. Adott a következő file. Hogyan alakítja át ezt a következő sedscrip?
`s\[([a-zA-Z]*) \([0-9]*\) / \2 \1/`

Éva	119
Marcsi	123
Edit	187
Kinga	141
Andi	193

8. IDŐZÍTETT PARANCSONK HASZNÁLATA

sleep [number]

Az argumentumban szereplő számnak megfelelő másodpercig felfüggeszti egy process futását.

sleep 10 10 másodperc időtartamig nem fogjuk visszkapni a promptot
(sleep 60; echo 'ebédidő!')& 60 másodperc múlva az ebédidő! string jelenik meg a monitoron.

at időpont

Segítségével lehetővé válik process-ek meghatározott időpontban történő elindítása. A parancs kiadása után egy másodlagos promptot kapunk, itt írhatjuk be a kiadandó parancsokat, soronként egyet-egyet. A parancsok begépelése után egy új sorban kiadott ^D-t kell begépelnünk. A parancs időpontként többféle formátumot is elfogad, a parancs eredményét levélben küldi el a usernek.

at 12 00	a továbbiakban felsorolt parancsok 12:00-kor lesznek végrehajtva
at 8:15am Jan 31	a továbbiakban felsorolt parancsok Jan. 31-én délelőtt 8:15- kor lesznek végrehajtva
at now + 1 day	a továbbiakban felsorolt parancsok a következő napon ugyanebben az időben lesznek végrehajtva

crontab -el [user]

Időzített, periodikusan ismétlődő parancsok kiadása. Az egyes userek rendelkezhetnek un. crontab file-lal (a /var/spool/cron/crontabs könyvtárban) amelyben meghatározott időpontokat és kiadandó parancsokat írhatnak elő.

- l saját crontab fájlunk listázása
- e saját crontab fájlunk editálása. (Előtte az EDITOR változot vi-re kell állítani !)

A crontab file bejegyzései 6 mezőből állnak. Az első 5 mező számot vagy speciális értelmezésű karaktereket tartalmaz, a 6. mező a parancsot írja le.

Az első 5 mező rendre a következőket adja meg:

perc	0-59
óra	0-60
a hónap napja	0-31
hónap	0-12
a hét napja	0-6 (0=vasárnap)

A hónap és a hét napja vagylagosan értékelődik ki.

Az első 5 mezőben alkalmazható speciális értelmezésű karakterek:

*	minden számra illeszkedik
k,l,m	a felsorolt számokra illeszkedik
a-n	a megadott intervallumra illeszkedik

```
0 * * * * cat /usr/demo/SOUND/sounds/cowbell.au > /dev/audio
Csengetés minden egész órában.
```

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Hozzunk létre saját crontab file-t. Ellenőrizzük működését!
4. Adjunk ki egy parancsot, amely 1 perc múlva megnézi, hogy milyen file-ok vannak a /tmp direktory-ban!
5. Az at parancs segítségével határozzuk meg, hogy hány óra lesz 5 perc múlva.
6. Hozzunk létre egy olyan file-t a /tmp könyvtárban, amely percnyi pontossággal mindig az aktuáli időt (óra, perc) tartalmazza.

9. A KORN SHELL HASZNÁLATA

A Korn Shell a Bourne shell továbbfejlesztéseként jött létre, azzal majdnem teljesen felülről kompatibilis. A továbbfejlesztés eredményeképpen lehetővé teszi a parancs visszahívást és a visszahívott parancsok editálásával újabb parancsok létrehozását.

A Korn Shellben a továbbfejlesztés eredményeként álneveket (aliasokat) is használhatunk. Ez lehetővé teszi, hogy bizonyos parancsokra más néven is hivatkozassunk, így a shellünk interaktív felületét saját ízlésünknek megfelelően alakíthatjuk át.

SHELL VÁLTOZÓK, KÖRNYEZETI VÁLTOZÓK

A shellnek, mint általában a programozási nyelveknek változói vannak. Ezek a változók létrehozhatók, értékük lekérdezhető ill. módosítható. Nevüket általában nagybetűvel szokás írni.

A shellek által értelmezett változók két csoportra oszthatók. A egyik csoportot az úgynevezett shell változók alkotják, amelyekre az a jellemző, hogy a megadott shell számára hordoznak információt, valamint, hogy csak az adott shellben látszanak, innen újabb programot indítva már nem lesznek elérhetők, ahhoz hasonlóan, mint ahogy a programozási nyelvekben a lokális változók csak az adott modulban élnek. A shell változók értékének megadása ill. nevének módosítása a következők szerint történhet:

VÁLTOZÓ=érték

Fontos, hogy sem az egyenlőségjel előtt, sem pedig utána nem állhatnak fehér karakterek.

A shell változók értékének lekérdezése az

echo \$VÁLTOZÓ

ill. a

set

parancs segítségével történhet. Az előző csak a megadott változó értékét adja meg, az utóbbi pedig az adott shellben élő valamennyi shell változó értékét kiírja. A shell változók az

unset VÁLTOZÓ

parancs segítségével szüntethetők meg.

A változók másik csoportja a környezeti változók kategóriája, amelyekre az a jellemző, hogy általában nem a shell, hanem az innen indított programok számára hordoznak információt, valamint, hogy nemcsak az adott shellben látszanak, újabb programot indítva továbbra is érvényben maradnak, a programozási nyelvekben használt a globális változókhoz hasonlóan. A környezeti változók értékének megadása ill. nevének módosítása a következők szerint történhet:

VÁLTOZÓ=érték; export VÁLTOZÓ

vagy

export VÁLTOZÓ=érték

A megadásból látható, hogy minden környezeti változó egyúttal shell változóként is szerepelni fog.

A környezeti változók értékének lekérdezése az

echo \$VÁLTOZÓ

ill. az

env

parancs segítségével történhet. Az előző csak a megadott változó értékét adja meg, az utóbbi pedig az összes környezeti változó értékét kiírja. A környezeti változók a shell változókhoz hasonlóan az

unset VÁLTOZÓ

parancs segítségével szüntethetők meg.

AZ INICIALIZÁLÁS FOLYAMATA

Shellünk inicializálódása függ attól, hogy az illető shell a bejelentkezéskor elinduló shell (login shell) vagy egy további shell, pl. a command tool, shell tool shellje. Login shell esetén először végrehajtódik a /etc/profile rendszerszintű inicializáló fájl. Ezt csak a rendszeradminisztrátor módosíthatja, azok a bejegyzések szerepelhetnek benne, amelyek minden felhasználóra érvényesek.

Ennek lefutása után - ha egyáltalán létezik - le fog futni a shellt indító user home directory-jában található .profile nevű rejtett fájl. Ezt a felhasználó maga is írhatja (gyakran ő maga is hozza létre), és olyan beállításokat szerepeltethet benne, amelyek kizárólag az illető userre érvényesek. Ezen beállításokkal kiegészítheti illetve módosíthatja a /etc/profile fájlban szereplő beállításokat.

A továbbiakban harmadik lépésként az ENV környezeti változó értéke által meghatározott fájl (shell script) fog lefutni, amennyiben ennek a változónak van értéke és az egy fájl. Ez az érték lehet például a /etc/env vagy a \$HOME/.kshrc. Az

ENV változó értéke tipikusan a /etc/profile fájlban van beállítva. Ha a user ettől a beállítástól el kíván térni, akkor ezt a home directory-jában található .profile fájlban az ENV környezeti változó értékének módosításával teheti meg.

Nem login shell esetén csak a harmadik lépés történik meg, azaz az ENV környezeti változó által meghatározott fájl fog lefutni.

SPECIÁLIS JELENTÉSSSEL RENDELKEZŐ PARANCSONK IDÉZÉSE

Gyakran van szükség arra, hogy a shell számára speciális jelentést hordozó karaktereket (pl. *, ?, > stb) idézzük, a shell elől elrejtjük, vagyis megóvjuk attól, hogy a shell ezeket értelmezze, kifejtse és emiatt szándékunktól eltérően járjon el, illetve a megfelelő alkalmazás elől elvegye őket.

Idézésre a szimpla, a dupla és a visszafelé idézőjel, valamint a \ karakter használható, de ezek rendre más-más módon idéznek.

A \ karakter az utána következő karaktert idézi, a szimpla idézőjel pedig idézi az egész tartalmazott stringet. Hasonlóképpen működik a dupla idézőjel is, a különbség csak az, hogy az utóbbi a \$ karakterrel kivételt tesz, azt nem idézi. A visszafelé idézőjelek közé egy UNIX parancsot írhatunk be, ezt a parancsot a shell végrehajtja és standard outputját a hivatkozott idézőjeles kifejezés helyére helyettesíti be, és az ilyen módon kapott parancssornak megfelelően jár el.

ALIAS PARANCSONK

Aliasok használatával operációs rendszerünk kezelői felületét magunk számára komfortosabbá tehetjük. Az alias parancs segítségével egy úgynevezett alias vagy álnév táblát lehet feltölteni. Ennek bejegyzései rendre egyenlőséggel összekapcsolt string párok. A shell egy parancs végrehajtása előtt megvizsgálja, hogy a parancssor első szava szerepel-e a tábla valamelyik bejegyzésének bal oldalán, ha igen, akkor helyettesíti azt az illető bejegyzés egyenlőségjelének jobb oldalán levő stringgel.

Aliast a következőképpen definiálhatunk:

```
alias alias_parancs='már létező parancs'
```

Fontos, hogy sem az egyenlőségjel előtt, sem pedig utána nem állhatnak fehér karakterek. Az így definiált alias parancs csak abban a shellben él, amelyikben őt létrehozták. Ha arra van szükség, hogy az alias parancs érvényes legyen tetszőleges shellben is, akkor az alias parancsot valamelyik shell inicializáló fájlban kell elhelyezni. Az alias tábla listázása az

```
alias
```

parancs kiadásával, az alias parancs megszüntetése egy parancs erejéig az illető alias parancs elé tett \ karakter, véglegesen pedig az

```
unalias alias_parancs
```

segítségével érhető el.

A PARANCS TÍPUSOK PRECEDENCIÁJA

A UNIX-ban a legnagyobb precedenciával az alias parancsok rendelkeznek, azaz, ha a kiadott parancsra létezik alias definíció, akkor megtörténik a behelyettesítés és az így előálló parancs fog végrehajtódni.

A következő precedencia szintet az úgynevezett belső parancsok jelentik. Ezek az egyszerű, a shellbe beépített parancsok (pl. `cd`, `set`, `env` stb.). Az alias parancs létezésének vizsgálata és az esetleges behelyettesítés után a shell azt vizsgálja meg, hogy létezik-e a szóban forgó parancsra belső parancs. Ha létezik, akkor az végrehajtásra kerül, ha nem akkor a harmadik precedencia szint következik. Ez a külső parancsok szintje, ezek olyan parancsok, amelyek önálló fájlként léteznek (pl. `ls`, `cp`, `mv` stb.). Ha létezik megfelelő külső parancs akkor a shell azt végrehajtja, ha nem, akkor hibaüzenetet kapunk.

PARANCSVISSZAHÍVÁS

A Korn Shell a korábban kiadott parancsokat a parancsokat kiadó user home directory-jában levő `.sh_history` nevű rejtett fájlban őrzi (a `HISTFILE` változó ettől eltérő megadásával a user másképpen is rendelkezhet). Így ezeket az újabb bejelentkezéseken túlmenően még akkor is visszahívhatjuk, ha a rendszer újra bootol.

A korábban már kiadott parancsok visszahívására ill. editálására két stílus lehetséges: a `vi` vagy az `emacs` stílus. Ezek beállítása a

```
set -o vi
```

ill. a

```
set -o emacs
```

parancsok segítségével lehetséges. Ugyanez érhető el az

```
EDITOR=vi
```

ill. az

```
EDITOR=emacs
```

beállításokkal. Ezeket a beállításokat természetesen valamelyik shell inicializáló fájlban érdemes előírni.

Emacs stílusú parancsvisszahívás esetén a korábban kiadott parancsok között egy paranccsal visszafelé a `^P` (previous), előre a `^N` (next) billentyűkombinációkkal léphetünk. Egy visszahívott parancssorban balra a `^B` (back), jobbra a `^F` (forward) hatására mozoghatunk. A backspace billentyű a megszokott módon visszafelé töröl, míg előre a `^D` segítségével törölhetünk. A `^U` a teljes visszahívott sort törli.

A legutoljára kiadott parancsok listája a

history

parancs segítségével tekinthető meg. A rendszer defaultként 128 parancssort tárol el, de a user a HISTSIZE változó megadásával ettől eltérően is rendelkezhet.

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a ksh-hoz tartozó man page-et!
3. Hozzunk létre saját \$HOME/.profile file-t!. Ellenőrizzük a működését! Miket célszerű ebbe a file-ba beírni?
4. Hozzunk létre egy saját nevünkkel azonos shell változót! Írassuk ki az értékét! Tegyük környezeti változóvá!
5. Írassunk ki a képernyőre az echo paranccsal egy csillagot!
6. Értjük el, hogy a dir string begépelése után az ls -l hajtódjon végre!
7. Értjük el, hogy az rm, cp, mv parancsok mindig interaktív módon működjenek!
8. Bővítsük a PATH változót a home direktory-nkban levő bin könyvtárral! Mit kell tennünk, hogy ez a beállítás a legközelebbi bejelentkezésünkkor is érvényes legyen?
9. Hogyan érhető el, hogy az echo a napi dátumot adja meg?
10. Hogyan érhető el, hogy az echo a PATH változó értéket adja meg?
11. Hozzunk létre \$HOME/.kshrc file-t! Értjük el, hogy inicializálódáskor harmadik lépésként ez a file kerüljön kiértékelésre !

10. PROCESS KEZELŐ PARANCSONK

Háttérben futtatott process indítása a parancs végén elhelyezett & jellel történik: command &

filemgr &	a háttérben futtatva elindítunk egy file managert
ls -lR / > file1 &	az ls -l-nek megfelelő hosszú rekurzív lista íródik az aktuális könyvtárban levő file1 nevű fájlba háttérben futó process következményeként. A hibaüzenetek a képernyőre kerülnek.
ls -lR / > file1 2>/dev/null &	az ls -l-nek megfelelő hosszú rekurzív lista íródik az aktuális könyvtárban levő file1 nevű fájlba háttérben futó process következményeként. A hibaüzenetek eldobásra kerülnek.

ps [-efl]

Megjeleníti a futó processeket és jellemzőiket.

- e valamennyi user összes process-ének jellemzői megjelenítődnek
- f a userek számára általában használható és érthető hosszú lista
- l hosszú lista, több információt tartalmaz, mint a -f kapcsoló hatására létrejövő

ps -ef | grep user1 információt jelenít meg azokról a process-ekről, amelyeket a user1 indított el

kill [-l] [SIGNAL] [PID]

Segítségével signálok, jelzések küldhetők a futó process-eknek. Ezek hatására például egy process futása befejeződik, átmenetileg szünetel vagy újraindul. A process-ekre való hivatkozás a PID (Process Identification Number) segítségével történik. A signal számával vagy szimbolikus nevével egyaránt megadható. A fontosabb signálok a következők:

- TERM (15) a process futása befejeződik (terminate). Ez a default signal érték, azaz, ha nem adunk meg signált. akkor a TERM lesz érvényes
- KILL (9) a process futása befejeződik. Ez erőszakosabb beavatkozást jelent a TERM-nél, ezzel még a saját login shellünk is megállítható.
- STOP (23) a process futása átmenetileg fel lesz függesztve
- CONT (25) a STOP állapotban levő process futása folytatódik

-l megjeleníti a signálok számát és szimbolikus nevét

kill 255	a 255-ös process futása a TERM signálnak megfelelően befejeződik
kill -KILL 255	a 255-ös PID-del rendelkező process futása befejeződik
kill -STOP 255	a 255-ös PID-del rendelkező process futása átmenetileg fel lesz függesztve

nice [-number] command

Az elindított process prioritásának beállítását teszi lehetővé. A userok process-szeik default prioritásán csak ronthatnak, a rendszeradminisztrátor azonban javíthat is. Az egyes processek prioritása a nice index segítségével állítható be, nagyobb nice index-hez kisebb prioritás tartozik. A nice index default értéke háttérben futó process esetén 24, előtérben futó process esetén pedig 20. A nice parancs argumentumában megadott számmal ezeket az értékeket módosíthatjuk, úgy, hogy az ott megadott szám a nice index default értékhez hozzáadódik. Az így kialakuló számnak 0 és 39 közé kell esnie, egyébként az intervallum határoknak megfelelően csonkolódik. Amennyiben a nice parancs argumentumában nem adunk meg számot, akkor a nice index 10-zel fog növekedni.

nice -15 ls -lR / > file1 &	az ls -l parancsnak megfelelő process 39-es nice index-szel, azaz igen alacsony prioritással fog futni.
nice --24 ls -lR / > file1 &	az ls -l parancsnak megfelelő process 0-s nice index-szel, azaz igen magas prioritással fog futni. Ez azonban csak akkor van így, ha a parancsot a rendszeradminisztrátor adta ki, a userok nem csökkenthetik a nice indexet a default érték alá, ezért esetükben a process 24-es nice index-szel fut.
nice ls -lR / > file1 &	az ls -l parancsnak megfelelő process 34-es nice index-szel fog futni.

nohup command

Szemben a nohup nélküli esettel, a nohup parancs argumentumában szereplő process akkor is tovább fog futni, ha az őt elindító user kijelentkezett.

nohup ls -lR / > file1 &	az ls -l parancsnál megfelelő process akkor is fut tovább, ha a parancsot kiadó user kijelentkezett
--------------------------	---

Az egyes parancsokról ill. kapcsolóikról további, részletesebb információ található az Online Manualban!

FELADATOK:

1. Próbáljuk ki a fentieket ! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Próbáljuk meg lelőni saját login shellünket! Lelőhetjük-e a mások által elindított processek valamelyiket?
4. Adjuk ki az `ls -lR / <file 2<`hiba parancsot úgy, hogy a process a háttérben fusson!
5. Vizsgáljuk meg, hogy csakugyan megtörténik a listázás!
6. Az előző pont szerinti parancs kiadása után lépünk ki, majd jelentkezünk be! Folytatódik-e a listázás?
7. Az előző pont szerinti parancsot módosítsuk úgy, hogy kiadása után a process tovább fusson!
8. Az előző pont szerinti parancsot módosítsuk úgy, hogy kiadása után a process prioritása változzon!
9. Keressünk olyan grafikus alkalmazásokat, amelyek „kiváltják” a tanulmányozott parancsokat!

11. HÁLÓZATI PARANCSONK

telnet [host]

Lehetővé teszi a telnet protokollon alapuló terminál emuláció, használatával saját gépünk gyakorlatilag a távoli gép termináljává válik. A felhasználó a távoli gép eléréséhez egy teljes környezetet kap, önálló parancskészlettel. Használatához a usernek a távoli gépen általában accounttal kell rendelkeznie. Bejelentkezés után a távoli gép szolgáltatásai igénybe vehetővé válnak, ugyanakkor a telnet parancsai is használhatók.

A telnet használata folyamán kétféle üzemmód érhető el.

1. Input mód: Ebben az üzemmódban minden begépett karakter a távoli géphez továbbítódik (kivéve az ún. escape karaktert, amely parancs módba visz).
2. Parancs mód: Ebben az üzemmódban érhetőek el a telnet parancsai. Ezek segítségével adhatunk parancsokat a kliensnek. Többféle telnet parancskészlet létezik, a parancsok (help, open, close, quit stb.) használata általában nem okoz nehézséget.

A használat során az escape karakter ismerete nagyon fontos. Ez általában a ^], de gyakran az F10 vagy az AltF10. A kapcsolat felépülésekor a képernyőn megjelenítődik, ahhoz hogy a parancs módot elérhessük meg kell jegyeznünk.

ftp [host]

Lehetővé teszi, hogy egy user egy tetszőleges gépen dolgozva fájlokat másoljon le egy távoli, tetszőleges földrajzi helyen levő gépről saját gépére ill., hogy saját gépéről fájlokat másoljon fel egy távoli gépre, függetlenül a gépek operációs rendszereitől.

E szolgáltatás igénybevételekor biztonsági problémák is felmerülnek, a távoli géphez való hozzáféréshez általában a távoli gépen accountra van szükség, csak ennek segítségével lehet a távoli géphez hozzáférni, természetesen a hozzáférési jogosultságrendszer szabta korlátokon keresztül. Egy fájlhoz való hozzáférés az ftp szolgáltatás használatakor lemásolhatóságot jelent, a userok a távoli gépeken levő fájlokról másolatokat készíthetnek és ezzel dolgozhatnak.

Az ftp szolgáltatás igénybe vétele online módon történik, ez valójában a távoli gépre való bejelentkezést jelent, ennek megtörténte után van lehetőség a fájl transzferre.

Az ftp használatakor egy korlátozott belső parancskészlet érhető el, ennek segítségével van lehetőség a távoli és a helyi gép kezelésére ill. a fájl transzferre. A fontosabb parancsok a következők:

help	lista az elérhető parancsokról
get	fájl letöltése
put	fájl felvitele
cd	az akt. könyvtár beállítása a távoli gépen
lcd	az akt. könyvtár beállítása a helyi gépen
bye	kilépés
open	kapcsolat felépítése
close	kapcsolat bontása
bin	bináris üzemmódú átvitel
ascii	ascii üzemmódú átvitel
status	az üzemmód jellemzőinek megjelenítése

Az ftp az egymással kapcsolatban álló két gép általában eltérő karakterábrázolása miatt karakterkonverziót hajt végre. Erre csak szöveges állományok esetén van szükség, bináris állományok esetén például azonban ez a konverzió tönkre is teheti az állományokat.

Ezért bizonyos fájlokat nem a default ascii, hanem az ún. binary (image) üzemmódban kell továbbítani. Egy adott fájl esetén az átviteli mód (ill. a fájl visszaállítási módjának) meghatározásához az alábbi táblázat nyújt segítséget.

fájlnev kiterjesztés	átviteli mód	visszaállítás	megjegyzés
.TXT, .txt	ascii	-	dokumentum fájl
.DOC, .doc	ascii	-	dokumentum fájl
.PS, .ps	ascii	-	postscript fájl
.Z	bin	uncompress	UNIX töm. fájl
.ZIP, .zip	bin	pkzip, pkunzip	töm. fájl
.gz	bin	gunzip	UNIX töm. fájl
.ZOO, .zoo	bin	zoo	zoo formátum
.Shar, .shar	ascii	unshar	UNIX Shell Arch.
.tar	bin	bin	UNIX Tape Arch.
.arj	bin	arj	DOS töm. file
.uu	ascii	uuencode/uudecode	UNIX spec. form.
.jpg	bin	-	JPEG graf. form.
.gif	bin	-	GIF formátum
.lhz	bin	lharc	töm. file

rlogin -l user host

Bejelentkezés távoli gépre, a távoli gépen levő account vagy user ekvivalencia alapján. (User ekvivalencia felhasználása esetén nincs szükség jelszóra!)

rsh távoli_host [-l user] parancs

A távoli gépen egy parancs (nem interaktív) végrehajtása. (A működés a user ekvivalencián alapszik, jelszó nem szükséges.)

rcp [-r] user1@host1:file1 user2@host2:file2

File másolás távoli gépek között. (A működés a user ekvivalencián alapszik, jelszó nem szükséges.)

User ekvivalencia: Távoli userek ill. jogosultságaik lokális usereknek ill. jogosultságaiknak vannak megfeleltetve.

User ekvivalenciák megvalósítása:

Rendszer szint: /etc/hosts.equiv konfigurációs file – rendszeradminisztrátori lehetőség.

User szint: \$HOME/.rhosts file

Bejegyzés	Jelentés
hostname	A konfigurálást végző user nevével azonos nevű távoli (a hostname-mel azonosított gépen accounttal rendelkező) user ekvivalens.
hostname username	A megadott nevű távoli user (a hostname-mel azonosított gépen accounttal rendelkező) user ekvivalens.

FELADATOK:

1. Próbáljuk ki a fentieket! Jegyezzük le tapasztalatainkat!
2. Tanulmányozzuk a fenti parancsok man page-eit!
3. Tanulmányozzuk a tenet parancs két üzemmódját!
4. Tanulmányozzuk az ftp parancs belső parancsait!
5. A szomszédunk gépéről másoljuk át az ftp parancs felhasználásával a /etc/passwd file-t bin és ascii üzemmódban? Látunk-e különbséget? Miért?
6. Szomszédunkkal együttműködve hozzunk létre user ekvivalenciákat!
7. Az rlogin, rsh és rcp parancsok közül melyik működik user ekvivalencia hiányában is?

8. Hozzunk létre a távoli gép /tmp könyvtárában egy fület!
9. A távoli gépen létrehozott file-ba írjunk egy tetszőleges szöveget az rsh és a cat parancs felhasználásával!
10. Indítsunk el a távoli gépen egy grafikus alkalmazást!