

Információs rendszerek üzemeltetése – Linux admin –

BME TMIT

2017

	Oldalszám
0. Bevezetés	2
0.1. De most komolyan!	2
0.2. Fontos tudnivalók	2
0.3. Végezetül...	3
0.4. A rendszer elindítása	4
1. A rendszer indítása után...	5
1.1. Feladatok	5
2. Az Apache 2 webkiszolgáló	7
2.1. Feladatok	7
3. Linux héjprogramozás	8
3.1. Feladatok	8
4. A monit	11
4.1. Feladatok	11
5. Appendix	13

0 — Bevezetés

Mindenki kész?
Igenis kapitány!
Hangosabban!
Igenis kapitány!

ÓÓÓÓÓÓÓÓÓÓÓÓÓÓÓÓÓÓ!
Ki lakik odalent, kit rejt a víz?

Spongyabob Kockanadrág

De most komolyan!

Ez a labormérés négy témakörön és rengeteg feladaton keresztül próbál képet adni a Linux adminisztrátorok világáról. De nem csak képet szeretne adni, hanem lehetőséget adni picit elmerülni ebben a furcsa világban. Éppen ezért ez a labormérés jelentős interaktivitást kíván meg: tipikusan minden feladatnál a Linux-os man oldalakat, Internetes keresőt kell használni. A feladatok után megadott támogató megjegyzések nem a megoldások, csak iránymutató segítségek – nem elég ezeket begépelni.

Fontos tudnivalók

A segédletben szereplő összes feladat tökéletes megoldása nem kötelező, de minden témakörből legalább 40%-ot el kell érni a labor sikeres teljesítéséhez. A labor sikeres elvégzésébe csak a ♠ szimbólummal jelölt feladatok számítódnak bele, a többi feladat amolyan ráhangolódás, illetve hasznos segítséget jelent a jelölt feladatok megoldásához. Így, természetesen, a nem jelölt feladatok megoldása nem kötelező, de hasznos lehet. Az opcionálisként jelölt feladatok megoldása sem kötelező, de ezen feladatok beleszámítanak a végső értékelésbe, mégpedig oly módon, hogy pontszámuk hozzáadódik azon témakör pontszámához, amelyből nem sikerült 40%-ot elérni.

Amennyiben egy feladat „ábécézve” van akkor az *A*, *B*, *C*, *D* részfeladatok közül nem kell mindet megoldani, csak azt, amelynek a kódja az Ön NEPTUN kódjából származtatható. Ehhez csak a NEPTUN kód karaktereinek a *ASCII* kódját (lásd 1. ábra) kell összegezni, majd venni a négyvel vett maradékát. Amennyiben a maradék *i*) nulla, akkor Önnek az *A* jelű feladatot, amennyiben *ii*) egy, akkor a *B* jelű feladatot, amennyiben *iii*) kettő, akkor a *C* jelű feladatot, és végezetül amennyiben *iv*) három, akkor a *D* jelű feladatot kell megoldani.

0	NUL	16	DLE	32	SPC	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

1. ábra. ASCII kódok

A feladatok sikeres megoldását egy (bash) héjprogram a labor végén automatikusan ellenőrzi, ennek elindításához mérésvezetői segédlet szükséges. A program nem csak ellenőrzi és értékeli a feladatokat, hanem magát a labor elvégzését bizonyító ellenőrző kóddal ellátott jegyzőkönyvet is legenerálja (és majd ezt a fájlt kell elküldeni az iru.bme@gmail.com címre). Sajnos egy „buta” program az emberi szemhez képest kevésbé elnéző, ezért a következő szabályok betartása gyakorlatilag elengedhetetlen a labor sikeres elvégzéséhez:

- Mindig használják a feladatokban megadott felhasználói neveket és jelszavakat! A jelszó az esetek döntő többségében az Önök NEPTUN kódja lesz, ezt írják mindig ugyanabban a formátumban (azaz ha a NEPTUN kód „al-mafa”, akkor az használható „AlMafA”-ként is, de akkor végig csak ebben a formában).
- Amennyiben futtatható héjprogramot, szkriptet kell írni, akkor annak nevét származtassa a feladat sorszámából az alábbi módon:
7.4.A feladat → 7_4.sh,
azaz a feladat sorszámában szereplő pontokat egyszerűen cserélje le alulvonásokra és illessze a végére az *sh* végződést. A programokat másolja minden esetben a *laboruser* felhasználó *bin* könyvtárába (*/home/laboruser/bin*). Amennyiben ez a könyvtár nem létezik, akkor hozza létre. Ne feledjen futtatási jogokat adni a fájlnak!
- Minden esetben pontosan kövesse a megadott specifikációkat!

Végezetül...

Eredményes labort és sok sikert kívánjunk Önöknek!

A rendszer elindítása

A VM elindítása

VMWare playerben indítsa el a virtuális gépet! Amennyiben az feldob egy ablakot, melyben a VM eredetére kíváncsi, válassza az „I copied it” opciót! A gazda rendszerbe visszatérni a Ctrl+Alt kombinációval lehet.

Hozzáférési adatok

A virtuális gépre két felhasználó áll rendelkezésre:

felhasználói név: **root**

jelszó: **irulabor**

felhasználói név: **laboruser**

jelszó: **laboruser**

A virtuális gépre *laboruser*ként jelentkezzen be!



1 — A rendszer indítása után...

Feladatok

1.1. feladat: Nézze meg, hogy a rendszeren mely szolgáltatások várnak hálózati kapcsolatokra (figyelnek adott porton), és azok mire valók!

`man netstat, netstat -l`

1.2. feladat: Nézze meg, mi a futtatott Debian rendszer kódneve!

`lsb_release -da`

1.3. feladat: Írassa ki a telepített csomagokat! Telepítve van a Midnight Commander fájlkezelő a rendszerben?

`dpkg -l`

♠ **1.4. feladat:** A céges policy-k szerint a távoli bejelentkezést biztosító szolgáltatásoknak (ssh) csak a lokális hálózatról kell elérhetőnek lenniük. Módosítsa a tűzfalszabályokat ennek megfelelően!

`iptables -L INPUT, iptables -A INPUT ...`

♠ **1.5. feladat:** A céges policy-k szerint a gépnek nem szabad válaszolnia a pingelésre. Módosítsa ennek megfelelően a tűzfal szabályait!

♠ **1.6. feladat:** Adjon hozzá a rendszerhez egy új felhasználót, akinek a neve legyen „mekkelek”, jelszava pedig az Ön NEPTUN kódja! Nézze meg, milyen új bejegyzés született az `/etc/passwd` fájlban!

`adduser`

♠ **1.7. feladat:** Tegye lehetővé az új felhasználó számára, hogy rendszergazda jogokkal futtathasson minden programot!

`visudo`

♠ **1.8. feladat:** A biztonság érdekében tiltsa le a `root` felhasználó SSH-n történő bejelentkezésének jogát!

`mc, /etc/ssh/sshd_config, /etc/init.d/ssh restart`

1.9. feladat: Valósítson meg RSA kulcsokkal működő autentikációt a fizikai és a virtuális gép között a saját, újonnan létrehozott (mekkelek) felhasználó számára! Ehhez az `ssh-keygen` program segítségével hozzon létre egy új publikus-privát kulcspárt. A létrehozott kulcspár privát részét másolja át a gazdagépre, majd a `puttygen` program segítségével konvertálja át a `PuTTY` számára emészthető formátumba. Végezetül a virtuális gépen a

```
1 $ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

parancs kiadásával engedélyezze Mekk Elek RSA bejelentkezését! Tesztelje le az új lehetőséget!

♠ **1.10. feladat:** Telepítse a MySQL-t a guest gépre! Az adminisztrátor jelszava az egyszerűség kedvéért legyen „root”! Természetesen éles környezetben nehezen kitalálható jelszavakat kell használni. Nézze meg, hogy a szerver melyik verzióját sikerült installálnia!

```
apt-get install
```

1.11. feladat: Vizsgálja meg a MySQL szerver beállításait (*/etc/mysql*)! Derítse ki, hogy milyen porton figyel a MySQL szerver, és hol tárolja az adatbázisokat.

1.12. feladat: A *netstat* parancs segítségével állapítsa meg, hogy fogad-e kéréseket a MySQL szerver, és ezt TCP vagy UDP protokollon teszi-e, továbbá hogy elérhető-e a MySQL szolgáltatás a képen kívülről, tehát a helyi hálózatról, vagy bármilyen IP címről!

```
netstat
```

1.13. feladat: Jelentkezzen be a telepítéskor megadott jelszóval rendszergazdaként a MySQL parancssorába, és listázza ki a adatbázisokat.

```
mysql -u root -p, SHOW DATABASES;
```

♠ **1.14. feladat:** A */root/students.sql* fájlban található exportált adatbázist töltsse fel a MySQL szerverre.

1.15. feladat: SQL parancsok segítségével ismerje meg az egyszerű mintaadatbázist! Találja ki, mi lehet a célja az adatbázisnak, és melyik táblában mit tárol!

```
SHOW DATABASES;, DESCRIBE students.students;
```

♠ **1.16. feladat (opcionális):** Töltsön fel táblánként legalább egy-egy új rekordot az adatbázisba, amelyek megfelelnek a sémáknak, és a táblák összefüggéseinek is. Az új hallgató neve legyen „Mekk Elek” az Ön NEPTUN kódjával. Mekk Elek 1974. április 1-én született.

```
SELECT, INSERT
```

2 — Az Apache 2 webkiszolgáló

Feladatok

♠ **2.1. feladat:** Telepítse az `apache2` csomagot a beépített csomagkezelőn keresztül!

```
1 $ apt-cache search apache2
2 $ apt-get install apache2
```

2.2. feladat: Ellenőrizze, hogy a webkiszolgáló beállítása lehetővé teszi-e a 80-as porton történő „hallgatózást”. Melyik fájl tartalmazza ezt a beállítást? Milyen paranccsal tudja ezt ellenőrizni böngésző nélkül?

♠ **2.3. feladat:** Az Apache 2 dokumentációja és az alap sablon alapján állítson be egy virtuális kiszolgálót, mely az `irulabor.vmware` domén névre töltődik be. A kiszolgáló által visszaküldött html oldalak kódja megtalálható és letölthető a <https://github.com/ng201/iru> címen. Ne feledje el aktiválni az elkészült konfigurációt!

Az `/etc/hosts` fájlban készítsen el egy bejegyzést, hogy a virtuális gép ismerje a `irulabor.vmware` nevet.

♠ **2.4. feladat:** A fizikai gépen (tehát nem a virtuális linuxon!) töltsse be a böngészőben a <http://irulabor.vmware/vedett> címet. Mint láthatja, a `vedett` mappában található fájlok jelenleg elérhetőek a fizikai gépről is. A feladat az, hogy a `vedett` mappa tartalma csak a virtuális gépről legyen elérhető a webszerverbe épített IP korlátozás segítségével.

♠ **2.5. feladat:** A feladat az, hogy a védett mappa tartalma csak felhasználónév és jelszó segítségével legyen elérhető a webszerverbe épített korlátozás segítségével. A felhasználókat és a hozzájuk tartozó jelszavakat egy külön fájl tartalmazza (`/etc/apache2/conf.d/.htpasswd`). A felhasználók között, természetesen, legyen ott „mekkelek”, akinek legyen a jelszava az Ön NEPTUN kódja!

♠ **2.6. feladat:** A feladat az, hogy a `nagyonvedett` mappa tartalma csak felhasználónév és jelszó segítségével legyen elérhető a webszerverbe épített korlátozás segítségével. A felhasználók a rendszer beépített felhasználói. Kíséreljen meg belépni a fenti címen.

`mod_authnz_external`, `pwauth`

2.7. feladat: A weboldal tartalmaz egy `nyilvanos` elnevezésű mappát is, benne egy `.htaccess` fájljal. Mire alkalmas ez a fájl?

♠ **2.8. feladat:** Az alapértelmezett webkiszolgáló konfigurációban ennek a fájlnak a használata tiltva van. Milyen konfigurációs direktívával tudjuk mégis engedélyezni, és ezt hol kell megadni? A kérdés a *nyilvanos* mappára vonatkozik, csak ott akarjuk engedélyezni a *.htaccess* használatát.

AllowOverride

♠ **2.9. feladat:** A *nyilvanos* mappára szeretné bekapcsolni a webszerver automatikus listázó funkcióját. Mit és milyen formátumban kell ehhez beírni a *.htaccess* fájlba?

+Indexes

3 — Linux héjprogramozás

Feladatok

♠ **3.1.A és B feladat:** Írassa ki a számítógép processzorának vendor id-jét! Az elkészített *bash* héjprogram (*/home/laboruser/bin/3_1.sh*) csak a vendor id-t írja ki a standard kimenetre, semmi mást!

♠ **3.1.C és D feladat:** Írassa ki a számítógép processzorának frekvenciáját! Az elkészített *bash* héjprogram (*/home/laboruser/bin/3_1.sh*) csak a processzor névleges frekvenciáját (mértékegység nélkül) írja ki a standard kimenetre, semmi mást!

♠ **3.2. feladat:** Írjon egy *bash* scriptet (*/home/laboruser/bin/3_2.sh*), amely megszámlolja a bemenetként kapott fájlban az üres sorokat!

```
1 $ cat file_3_2 | ./3_2.sh
2 12
```

♠ **3.3.A és C feladat:** Írjon egy *bash* scriptet *3_3.sh* néven, amely paraméterként egy pozitív egész számot kapva kilistázza a standard bemenete kapott öt oszlopból álló adatfolyam azon sorait, amelyek a parancssori paraméterként megadott számmal osztható számmal kezdődnek, és teszi mindezt úgy, hogy közben felcseréli a második és az ötödik oszlopok tartalmát.

```
1 $ cat input.txt
2 1 a1 b c1 d
3 2 a2 b c2 d
4 3 a3 b c3 d
5 4 a4 b c4 d
6 5 a5 b c5 d
7 $ cat input.txt | ./3_3.sh 2
8 2 d b c2 a2
```



```

9 4 d b c4 a4
10 $ cat input.txt | ./3_3.sh 3
11 3 d b c3 a3

```

♠ **3.3.B és D feladat:** Írjon egy *bash* scriptet *3_3.sh* néven, amely paraméterként egy pozitív egész számot kapva kilistázza a standard bemenete kapott öt oszlopból álló adatfolyam azon sorait, amelyek a parancssori paraméterként megadott számmal osztható számmal kezdődnek, és teszi mindezt úgy, hogy közben összefűzi a negyedik és az ötödik oszlopok tartalmát.

```

1 $ cat input.txt
2 1 a1 b c1 d
3 2 a2 b c2 d
4 3 a3 b c3 d
5 4 a4 b c4 d
6 5 a5 b c5 d
7 $ cat input.txt | ./3_3.sh 2
8 2 a2 b c2d
9 4 a4 b c4d
10 $ cat input.txt | ./3_3.sh 3
11 3 a3 b c3d

```

♠ **3.4. feladat:** Írjon egy *bash* héjprogramot (*/home/laboruser/bin/3_4.sh*), amely futtatáskor soronként kiírja

1. a futtató felhasználó nevét,
2. az aktuális dátumot (éééé. hh. nn. formátumban),
3. a bejelentkezett felhasználókat (mindegyiket egyszer és külön sorba) és
4. a felhasználó által épp futtatott *3_4.sh* script pid-jét.

♠ **3.5. feladat (opcionális):** Írjon egy *bash* scriptet *3_5.sh* néven, amely a szabványos CSV formátumból konvertál a magyar Excel számára értelmezhető CSV formátumába, azaz az oszlopokat elválasztó vesszőket lecseréli pontosvesszőkre. Vigyázzon, hogy csak az oszlopokat elválasztó vesszőket módosítsa a program! A konvertálandó állományt a standard bemeneten kell fogadnia a programnak. A feladat megoldása során, az egyszerűség kedvéért, feltételezheti, hogy az oszlopok száma nem haladja meg a hármat, illetve, hogy az oszlopok csak az angol ábécé kis- és nagybetűit, valamint a `;;?!()<>[]` karakterek tartalmazzák.

♠ **3.6. feladat:** Írjon egy *bash* scriptet *3_6.sh* néven, amely parancssori paraméterként megkapja két létező fájl nevét, és megállapítja, hogy hány különböző sor található bennük. A script futási eredményeként két sor jelenik meg a szabványos kimeneten: az első sor tartalmazza azoknak a soroknak a számát, amelyek benne vannak az elsőként megadott fájlban, de nincsenek benne a másodikban, a második sor pedig ugyanezt az adatot tartalmazza vica-versa.

diff

A 3.7-es feladatok teszteléséhez az alábbi formátumú, csoportok és értékpárok összerendelését tartalmazó fájl használható:

```
1 $ cat file_3_7
2 Item1,2,200
3 Item2,3,500
4 Item3,1,900
5 Item2,2,800
6 Item1,2,600
```

A fájl csak nullánál nem kisebb számokat tartalmaz!

♠ **3.7.A feladat:** Írjon egy *bash* scriptet *3_7.sh* néven, amely összegzi az egyes csoportokba tartozó számok szorzatának az összegét. A script futását az alábbi kódrészlet szemlélteti:

```
1 $ cat file_3_7 | ./3_7.sh
2 Item1: 1600
3 Item2: 3100
4 Item3: 900
```

♠ **3.7.B feladat:** Írjon egy *bash* scriptet *3_7.sh* néven, amely kiírja minden csoport esetén a harmadik oszlopban található legkisebb elemet. A kimenet formátuma kövesse az alábbi sémát:

```
1 $ cat file_3_7 | ./3_7.sh
2 Item1: 200
3 Item2: 500
4 Item3: 900
```

♠ **3.7.C feladat:** Írjon egy *bash* scriptet *3_7.sh* néven, amely kiírja minden csoport első előfordulását. Azaz a működése és kimeneti formátuma legyen:

```
1 $ cat file_3_7 | ./3_7.sh
2 Item1: 2, 200
3 Item2: 3, 500
4 Item3: 1, 900
```

♠ **3.7.D feladat:** Írjon egy *bash* scriptet *3_7.sh* néven, amely megszámlolja majd kiírja az egyes csoportok előfordulásának a számát, azaz:

```
1 $ cat file_3_7 | ./3_7.sh
2 Item1: 2
3 Item2: 2
4 Item3: 1
```

♠ **3.8 feladat (opcionális):** Írjon egy *bash* scriptet *3_8.sh* néven, amely egy három oszlopból álló (az oszlopokat minden esetben egy pontosvessző választja el) állomány második oszlopában szereplő 0 és 1 közötti számoknak 0.1 széles intervallumokba osztja, majd kiírja, egymástól vesszővel elválasztva, az egyes intervallumok gyakoriságát.

4 — A monit

Feladatok

4.1. feladat: Határozza meg a gépen futó leginkább processzor igényes alkalmazást!

`top`

♠ **4.2. feladat:** Töltse le a virtuális gépen található operációs rendszer verziójának megfelelő *monit* programot a `/root/bin` könyvtárba (A labor során használja a tantárgy honlapján megtalálható verziót!). A program elérési útvonala legyen `/root/bin/monit/bin`! Indítsa el, majd jelentkezzen be a program webes felületén a `admin/monit` alapértelmezett felhasználónév/jelszó kombinációval. Ezen adatokat a feladatok során ne változtassa meg! Ügyeljen arra, hogy a *monit* parancssori paraméterek megadása nélkül is indítható legyen!

`https://mmonit.com/monit/`

♠ **4.3. feladat:** Módosítsa a *monit* konfigurációs állományát, hogy jelezzen, ha a memóriahasználat túllépi a 25%-ot. A problémáról a program küldjön e-mailt a `laboruser@irulabor.irulabor` címre. Az e-mail tartalmazza az alapértelmezett mezőkön kívül az Ön NEPTUN kódját a

```
1 Neptun: NEPTUN
```

formátumban rögtön a hostot megadó sor után! Ellenőrizze, hogy az e-mail megérkezett!

`cat /var/mail/laboruser`

4.4. feladat: Töltse le a virtuális gép Linux disztribúciójának és verziójának megfelelő *stress* és *cpulimit* programokat. A letöltött programok segítségével írjon egy bash scriptet, amely 25%-ig leterheli a CPU-t.

```
stress -c 1 & cpulimit -p $( pidof -o $! stress ) -l 25
```

4.5. feladat: Módosítsa a *monit* konfigurációs állományát úgy, hogy a monitorozó program jelezzen, ha olyan futó alkalmazást észlel, amely legalább 33%-ig leterheli a CPU-t. A problémáról a *laboruser@irulabor.irulabor* e-mail címre küldött levél tartalmazza, az alapértelmezett mezőkön kívül, az Ön NEPTUN kódját a

1 Neptun: NEPTUN

formátumban rögtön a munkaállomást megadó sor után! Ellenőrizze, hogy az e-mail megérkezett!

♠ **4.6. feladat:** Módosítsa a *monit* konfigurációs állományát, hogy a monitorozó program termináljon minden olyan futó alkalmazást, amely legalább 33%-ig leterheli a CPU-t. Tesztelje le, hogy működik-e az Ön által írt feltétel!

5 — Appendix

Linux héjak és beépített parancsok

Néhány közkedvelt Linux héj:

<i>bash</i>	a GNU <i>Bourne-Again Shell</i> je
<i>ksh</i>	a <i>Korn shell</i> , az eredeti vagy egy klonja
<i>pdksh</i>	a <i>Public Domain Korn</i> héj
<i>sh</i>	az eredeti <i>Bourne shell</i>
<i>zsh</i>	a <i>Z shell</i>

Fontosabb beépített parancsok:

<i>.</i>	beolvassa és végrehajtja az aktuális fájlt
<i>cd</i>	megváltoztatja az aktuális könyvtárat
<i>eval</i>	végrehajtja a szöveggént megadott héjprogramot
<i>exit</i>	kilép a héjprogramból
<i>read</i>	beolvas egy értéket az inputról
<i>test</i>	kiértékeli a paraméterként megadott kifejezést
<i>unset</i>	„törli” a héj egy változóját vagy függvényét

A mindennapi életben hasznosnak bizonyult utasítások:

<i>basename</i>	kiírja az útvonal utolsó komponensét opcionálisan a suffix elhagyásával
<i>dirname</i>	kiírja az útvonalat az utolsó elemét leszámítva
<i>id</i>	kiírja a felhasználó azonosítóját és nevét
<i>date</i>	kiírja az aktuális dátumot és időt a paraméterekben specifikált formátumban
<i>who</i>	a bejelentkezett felhasználók kilistázása
<i>stty</i>	az aktuális terminál beállításainak a kezelése

Szövegkezelés

<i>awk</i>	szöveges állományok feldolgozására reguláris kifejezések segítségével
<i>cat</i>	fájlkonkatenáció
<i>cmp</i>	egyszerű program fájlok összehasonlítására
<i>dd</i>	blokk szintű adatmozgatás
<i>echo</i>	argumentum kiírása a standard kimenetre
<i>egrep</i>	kibővített <i>grep</i> , amely az ERE típusú reguláris kifejezéseket használja
<i>expand</i>	tabulátorok szóköz karakterekre cseréje
<i>fgrep</i>	gyors <i>grep</i>
<i>grep</i>	<i>g/re/p</i> :)
<i>less</i>	hosszú fájlok lapozása („Less is more.”)
<i>more</i>	az eredeti BSD Unix lapozó program
<i>sed</i>	karakterfolyamok módosítása
<i>sort</i>	szöveges fájlok rendezése
<i>spell</i>	helyesírás-ellenőrző
<i>tee</i>	a sztenderd bemenetet a sztenderd kimenetre és a megadott fájlba másolja
<i>uniq</i>	duplikált sorokat eltávolítása rendezett bemenetből
<i>wc</i>	sorok, szavak és a karakterek megszámlálása a bemenetben

Fájlkezelés

<i>chgrp</i>	fájlok és könyvtárak csoportjának a megváltoztatása
<i>chmod</i>	fájlok és könyvtárak hozzáférési jogának a megváltoztatása
<i>chown</i>	fájlok és könyvtárak tulajdonosának a megváltoztatása
<i>cp</i>	fájlok és könyvtárak másolása
<i>df</i>	üres helyek mérete a háttértárolón
<i>diff</i>	fájlok összehasonlítása
<i>du</i>	a diszkek foglaltsági adatainak a megjelenítése
<i>gzip, gunzip</i>	tömörítő program és kitömörítő programok
<i>head</i>	fájlok első <i>n</i> sorának a listázása
<i>locate</i>	fájl keresése a neve alapján
<i>ls</i>	fájlok listázása
<i>md5sum</i>	ellenőrző összeg számolása MD5 algoritmussal
<i>mkdir</i>	új könyvtár készítése
<i>pwd</i>	aktuális könyvtár kiírása
<i>rm</i>	fájlok és könyvtárak törlése
<i>rmdir</i>	üres könyvtárak törlése
<i>sha1sum</i>	ellenőrző összeg számolása SHA1 algoritmussal
<i>tail</i>	fájlok utolsó <i>n</i> sorának a kiírása
<i>tar</i>	szalagarchiváló
<i>touch</i>	fájlok hozzáférési idejének a módosítása
<i>umask</i>	alapértelmezett hozzáférés beállítása fájlok számára

Folyamatkezelés

<i>fuser</i>	adott fájlt vagy szocketet használó folyamatok megkeresése
<i>kill</i>	jelzés küldés egy vagy több folyamatnak (tipikusan „kill” küldése, hogy fejezze be a futását)
<i>nice</i>	folyamatok prioritásának a megváltoztatása az elindításuk előtt
<i>ps</i>	információ a futó folyamatokról
<i>sleep</i>	a végrehajtás felfüggesztése a megadott időre
<i>top</i>	a leginkább CPU-igényes folyamatok listázása

Egyébb programok

<i>man</i>	utasítás, függvény, rendszerhívás, stb. manuáljának a listázása, a manuálból a q betű-vel lehet kilépni
<i>scp</i>	biztonságos távoli fájlmásolás
<i>ssh</i>	„secure shell”
<i>uptime</i>	megadja a legutóbbi bekapcsolás óta eltelt időt, illetve a rendszer terheltségi adatait