

Operációs rendszerek gyakorlat – összefoglaló

Az alapok

UNIX:

- 1970-es években jelent meg
- PDP-11 az első
- C-ben írták

Linux:

- UNIX átirata PC-kre
- Linus Torvalds 1991-ben kezdte el fejleszteni. Nyílt forráskód, egyre többen fejlesztették.
- A Linux kabalafigurája Tux, a pingvin.

kernel: rendszermag

disztribúció: a segédprogramok, a felhasználói programok és a kernel összessége

pl.: Ubuntu, Uhu, Debian, RedHat, Fedora, SUSE (több száz Linux disztribúció létezik)

Az **Ubuntu** a Debianon alapul. Fejlesztése mögött a Canonical áll. Első kiadása 2004 októberében jelent meg. Hathavonta jön ki új verziója. Verziószáma éé.hh alakú (pl.: 11.10, a 2011. októberi kiadásáé).

abszolút elérési út: közvetlen a gyökértől fejtjük ki, hol van az állomány

relatív elérési út: az éppen aktuális könyvtárhoz képest, hol van (A . [pont] az aktuális könyvtárt jelöli, a .. [pont pont] pedig az aktuális könyvtár szülőkönyvtárát.)

parancsértelmező (burok, héj, shell): az a program, ami egy felhasználó belépésekor aktivizálódik. Feladata, hogy a felhasználó által begépelte karaktersorozatokat szétbontsa, megpróbálja végrehajtható parancsként értelmezni, feldolgozza és kifejtse a parancsnak átadandó paramétereket, megkeresse az elindítandó programfájlt, és megpróbálja azt elindítani; a parancs végrehajtása után pedig visszaadja a vezérlést a felhasználónak, s kész legyen az újabb begépelte parancsok végrehajtására.

A UNIX rendszereken többféle shell program futtatható. A legrégebbi a Bourne (sh vagy bsh), C nyelv szintaxisát követő C-shell (csh), Korn shell (ksh) és a bash - GNU Bourne-Again Shell.

A **bash shell** maga egy interaktív parancsértelmező és programozási nyelv, amely a következő funkciókat tartalmazza: shell parancsok, függvények, feltételek és aritmetikai kifejezések kiértékelése, különleges parancsok, processz vezérlés, parancs helyettesítés, állománynév helyettesítés, bemenet és kimenet átirányítása, helyettesítő nevek alkalmazása, ~ (tilde) helyettesítés, paraméter helyettesítés és a parancssor szerkesztése. Ha a bash shell bejelentkezéskor hívja meg a rendszer, az végrehajtja a `/etc/profile` shell állomány parancsait, majd sorban a `~/.bash_profile`, `~/.bash_login`, és `~/.profile` állományok parancsait. Kilépéskor végrehajtja a `~/.bash_logout` parancsait.

Ha a bash shell nem-bejelentkezéskori interaktív shellként fut, akkor megpróbálja a `~/.bashrc` konfigurációs állomány parancsait futtatni. Amennyiben a bash parancs után megadunk egy állománynevet, akkor az adott állományban felsorolt shell parancsok végrehajtnak.

A héj **prompt**tal jelzi, ha készen áll utasítások fogadására. Az alapértelmezett prompt függ a használt héjtól, mely bash shell-nél ebben a formában jelenik meg:

`<felhasználónév>@<hostnév domain név nélkül> <aktuális könyvtár> $ vagy #`

(A prompt utolsó karaktere `#` root felhasználó esetén, minden más felhasználó promptja pedig `$`-el végződik.)

Ha a shell további bevitelre vár, mielőtt végre tudja hajtani a parancsot, a `>` promptot kapjuk, például, ha egy parancsot több sorban adunk ki. A parancs végrehajtására kész promptot **PS1 promptnak**, míg a további bevitelre váró promptot **PS2 promptnak** nevezzük. Az alapértelmezett promptok megváltoztathatók, ha a PS1 illetve PS2 változóknak új értéket adunk. Hogy hogyan épül fel az aktuális prompt, megkapjuk a változó értékének képernyőre íratásával.

(A `/u` a felhasználói nevet, a `/h` a hostnevet, a `/w` az aktuális könyvtárt jelöli.)

A UNIX-hoz szállított kézikönyvek rendszerint két fő csoportra oszthatóak, nevezetesen a referencia kézikönyvekre (Reference Manual), és az összes egyébre. A referencia kézikönyvek nevüknek megfelelően az egyes parancsok, rendszerhívások stb. részletekbe menő ismertetését nyújtják, a többi kötet vagy valamilyen bevezető szintű, vagy haladóbb ismereteket ad a rendszer használatának egyes vonatkozásairól. Lehetőség van rá, hogy a kézikönyvekhez online módon férjük hozzá: A **man** parancs hatására (paraméternek a referencia kézikönyv valamelyik tételének nevét kell megadni, például `man cat`, vagy `man hosts`) a megnevezett UNIX parancs, rendszerhívás, vagy szubrutinra vonatkozó manuál oldalakat kilistázza a rendszer a képernyőre.

Fájl- és könyvtárkezelés

cd – a könyvtárváltás parancsa

A `cd` parancs kiadásakor a home könyvtárunkba jutunk

A `cd ..` hatására az aktuális könyvtár szülőkönyvtárába kerülünk

pwd – kiírja az aktuális könyvtárat

ls – a könyvtár tartalmát listázza

`ls -l` – részletes információkat ír ki az állományokról

`ls -a` – kilistázza a rejtett állományokat is, stb

rm – állomány törlése

`rm -i` – megerősítést kér a törlés előtt

touch – üres állomány létrehozása

mkdir – könyvtár létrehozása

rmdir – üres könyvtár törlése

cat – állomány tartalmát írja ki

cp – állomány másolása

`cp /honnan/mit /hová/milyen_néven`

`cp -r` – könyvtárak másolása

mv – állomány átnevezése vagy áthelyezése

`mv mit hová; mv mit mire`

chmod – fájlok és könyvtárak jogosultságait állíthatja be

`chmod u+x fájlnev` – az aktuális felhasználó végrehajtási jogot kap

`chmod -R` – rekurzívan megváltoztatja a könyvtárnak illetve tartalmának jogosultságait

Fájlok jogosultságai

- minden állománynak van tulajdonosa, csoportja és mindenki más
- mindezekhez tartozik olvasási (**read**), írási (**write**) és futtatási (**execute**) jog
- a fájl(ok) futtatásához **rx** kell és a mappa megnyitáshoz is szintén **rx** kell

Ha a jogosultság kötőjellel (-) kezdődik, akkor fájlról van szó, ha d betűvel, akkor könyvtárról, és ha l betűvel, akkor linkről van szó.

A többi 9 karaktert hármasával tagoljuk. Ezek jelentik a tulajdonosra, a csoportra és a mindenki másra vonatkozó jogosultságokat.

Jogosultság – <i>chmod</i> parancs				Jogosultság – <i>chmod</i> parancs			
Engedélyek	Tulajdonos	Csoport	Mindenki más	Engedélyek	Tulajdonos	Csoport	Mindenki más
Olvasás	r	r	r	Olvasás	4	4	4
Írás	w	w	w	Írás	2	2	2
Futtatás	x	x	x	Futtatás	1	1	1
Összegezve:	rwX	rwX	rwX	Összegezve:	7	7	7

UNIX fájlrendszer

Az UNIX fájlrendszer többelágazású fa, mivel a gyökérnek akárhány gyermeke lehet.

/	A hierarchikus könyvtárfa kiindulópontja (gyökér könyvtár)
/boot	A rendszer indításához szükséges állományok helye (grub, vmlinuz, stb)
/bin	A futtatható parancsok könyvtára – binaries
/sbin	A rendszergazda parancsai – superuser bin
/lib	Az induláshoz szükséges osztott rendszerkönyvtárak – libraries Továbbá tartalmazza a rendszerhez csatlakozható modulokat, meghajtóprogramokat
/dev	A rendszerhez csatlakozott, csatlakozható különleges állományok – devices
/etc	Beállítófájlok, helyi indító parancsok, jelszavak, hálózati-beállítók, etc. helye.
/home	Minden felhasználó saját könyvtára itt foglal helyet
/mnt	A felcsatolt (mountolt) perifériák könyvtára – mount
/proc	A rendszer-erőforrásokról tartalmaz információkat – process information
/root	A rendszer gazdájának könyvtára
/tmp	Ideiglenes adatok tárolására használt könyvtár – temp
/usr	A felhasználói programok tárhelye
/var	Változó adatokat tartalmazó állományok könyvtára

df – szabad terület számítása, partíciónként

df -h – emberek számára olvashatóbb formában írja ki

ln -s – soft linket hoz létre

ln -s *mire_mutat* *mutató_neve*

diff – két fájlt hasonlít össze

A vim

normál mód: ESC lenyomásával térhetünk vissza. Számos parancs csak normál módban működik.

insert mód: az i billentyű lenyomásával léphetünk be. Ekkor módosítani tudjuk a szöveget: beírás.

vizuális mód: a Shift+V kombinációval kijelöljük az első sort, majd a lefelé nyíllal még több sort tudunk kijelölni. Kijelölés után ezekre a sorokra vonatkoznak a parancsok.

parancs mód: normál módból a kettőspont (:) leütésével léphetünk be.

:q! – kilépés mentés NÉLKÜL, kérdés nélkül

:q – kilépés [ha nem módosítottuk a fájlt]

:w – mentés

:x – kilépés MENTÉSSEL, kérdés nélkül [vagy :wq! vagy ZZ]

h,j,k,l – balra, le, fel, jobbra

Ctrl+F, vagy **Ctrl+B** – ugrása előre, vagy hátra egy oldalt [PgUp vagy PgDown] pl.: beírod, hogy "10", utána PgDown gomb=10 oldalt ugrik lefelé

G – a fájl végére ugrik [vagy :\$]

gg – a fájl elejére ugrik

^ – az első nem whitespace karakteréhez ugrik

0 vagy **\$** – sor elejére vagy végére ugrani [Home vagy End]

w – egy szót előre [b – hátra]

{ – ugrás a bekezdés elejére **}** – bekezdés végére]

% – ha zárójelen állunk, átugrik a párjára [és fordítva]

fu – előre az első "u" betűig [soron belül]

Fu – vissza az utolsó "u" betűig [soron belül]

:300 – a 300. sorhoz ugrik [vagy "300G"]

20| – a 20. oszlophoz ugrik a soron belül

Ctrl+G – megmutatja, hogy módosított-e a fájl + hol vagy a fájlban belül

w – a következő szó elejére ugrik [b – az előző szó elejére ugrik]

***** – átugrik a következő szóra, ami egyezik azzal, amin álltál, mikor megnyomtad a *****-ot [# -al visszafelé ugyanez]

o – új sor nyitása a kurzor alatt [**O** – a kurzor felett]

x – kurzor alatti karakter törlése [3 karakter törlése pl.: 3x] [kurzor előtti karakter törlése: X]

dw – kurzortól a szó végéig törlés, beleértve a szóközt [5dw – ua., csak 5-ször.]

de – kurzortól a szó végéig törlés, NEM beleértve a szóközt [mármint az utána levő szóközt]

d\$ – kurzortól a sor végéig törlés

dd – sor törlése [2dd – 2 sor törlése pl.] [D – ua.]

%d – fájl tartalmának törlése

daw – kurzor pozíciótól függetlenül, ha a szón belül vagyunk, kitörli azt

d} – törlés a bekezdés végéig

c0 – a kurzor és a sor eleje közti szöveg átírása

cw – átírás a szó végéig [3cw – a következő 3 szó átírása] [a "c" a "d" "párja", de nem csak töröl, hanem i-be is rak utána]

r – a kurzor alatti karakter cseréje egy másikkal [replace]

s – a kurzor alatti karakter helyettesítése bármennyi karakterrel [substitute, i módba tesz]

3s – a következő három karakter helyettesítése bármennyi karakterrel [i módba tesz]

d/valami – törlés előre, amíg "valami"-ig nem érünk, "valami" megmarad

d?valami – ua., csak vissza, és a "valami" törlésre kerül

/valami – keresés előre [a "valami" szót]

?valami – keresés hátra

n – ugrás a következő találatra

N – ugrás az előző találatra

u – utolsó parancs visszavonása

U – az egész sort helyreállítja

Ctrl+R – ismét [mármint visszavonás ellentettje]

yy – az egész sor a vágólapra [vagy Y] [5yy – öt sor a vágólapra]

d5d – kivágja a kurzortól számított öt sort; (p-vel beillesztheted később azt a kívánt helyre)

p – vágólap tartalmának beillesztése a kurzor után, paste [P – elé]

yyp – aktuális sor másolása és beillesztése [vagy Yp]

y} – másolás a bekezdés végéig

gU – a kijelölésben szereplő szöveget átkonvertálja nagybetűssé [vagy U] [gu – kisbetűssé] [de lehet most is ötvözni pl.: gU10l – a következő 10 karaktert nagybetűssé alakítani]

A Midnight Commander

Ctrl+O – váltás az előtérprogram és a háttérprogram között

Esc+0 – kilépés

Insert nyomva tartása – több fájl kijelölése

F6 – átnevezés vagy áthelyezés

Ctrl+U – panelek felcserélése

cd / – gyökérkönyvtárba lépés

cd – home könyvtárba lépés

Esc 2x (OK) előtt – művelet félbehagyása

F9 – menüsorba lépés

Folyamatok

ps – futó folyamatok állapotának kiírása

ps ux – saját futó folyamataink

kill – egy folyamat kilövése azonosító alapján

killall – egy folyamat kilövése név alapján

bg – a háttérben folytatódik a legutóbb felfüggesztett folyamat
fg – az előtérben folytatódik a legutóbb felfüggesztett folyamat
folyamat háttérben futtatása – *parancs &*

Ismerkedés a bash parancsértelmezővel

- HOME könyvtár: a felhasználó home könyvtára
- a .bashrc fájl: a felhasználó konfigurációs állománya, amely a bash shellre vonatkozó beállításokat tartalmazza

Aliasok

Az alias parancs segítségével hosszabb vagy gyakran használt utasítást, utasítássorozatot tudunk le-
rövidíteni.

Az összes alias kilistázása az alias paranccsal történik.

Új alias **létrehozása**: alias *név*='utasítás'; **használat**a a *név* kiadásával történik.

Az így létrehozott alias csak az aktuális terminálablakban használható. El**mentés**éhez a ~/.bashrc
fájlban kell elhelyezni.

Alias **megszünt**etése: unalias *név* parancs kiadásával

Változók

Léteznek globális és lokális változók.

A globális változókat minden, a parancsértelmezővel indított program megkapja, és azok értékét
működése közben felhasználhatja. Szükség esetén meg is változtathatja, de ennek csak a saját kör-
nyezetében és az általa indított programokra lesz hatása.

A lokális változók értéke csak a parancsértelmezőn belül használható fel.

A parancsértelmező használata során a változókra a nevükkel hivatkozhatunk, a nevet azonban
(hogy megkülönböztethető legyen a parancssorban található argumentumoktól) a \$ karakter vezeti
be. A \$ karakter után közvetlenül, vagy kapcsos zárójelek közé zárva egyaránt állhat a változó neve.

Változó értékének lekérdezése: echo \$*név*.

echo – kiír egy szövegsort

set – az előre definiált változók listája

source – újratölti a .bashrc állományt; lehetővé teszi több forrásfájl használatát egyszerre

expr – egyszerűbb aritmetikai műveletek elvégzése bash alatt (A műveleti jelek között szóközt kell
hagyni!)

python – bonyolultabb aritmetikai műveletek elvégzése bash alatt (Az utolsó művelet eredményét
eltárolja az _ változóban.)

Előredefiniált változók

- HOME – a felhasználó saját home könyvtárának elérési útvonalát tartalmazza
- SHELL – az aktuális shell elérési útját tartalmazza
- USER – a felhasználói nevet tárolja
- PATH – elérési útvonalakat tartalmaz a parancsokhoz
- PS1 – az elsődleges prompt beállításait tartalmazza
- PS2 – a másodlagos prompt beállításait tartalmazza
- UID – a felhasználó egyéni azonosítóját tartalmazza

Szűrők

Input/output átirányítás

parancs > *fájl* – a standard output átirányítása egy fájlba (ha nem létezik, létrehozza, ha igen, felülírja)

parancs >> *fájl* – a kimenetet az eddigi fájlhoz fűzi hozzárendelődik

parancs < *fájl* – a standard input megváltoztatása a *fájl*-ra

parancs 2>*fájl* – a standard hibakimenet a *fájl*-ba kerül

parancs 2>/dev/null – hibaüzenetek elrejtése (mivel a /dev/null a rendszer „fekete lyuka”)

Lehetőség van egyszerre több átirányításra is:

pl.: *parancs* > *fájl* 2>&1 – a standard kimenetet és a standard hibát is a fájlba irányítja

A standard fájl-leírók

A Unix rendszer fájl-orientált szervezése a parancsok futtatásakor is megmutatkozik: minden egyes futó parancshoz hozzárendelődik három fájl:

- a standard input (stdin): 0; alapértelmezetten ez a billentyűzet
- a standard output (stdout): 1; alapértelmezetten ez a képernyő
- és a standard error (stderr): 2; alapértelmezetten ez a képernyő

A pipe használata

A pipe, melynek jele a | karakter, az egyik program kimenetét a másik program bemenetével köti össze.

Pl.: `ls -l /bin | more` – kilistázza a /bin könyvtárat, majd oldalanként jeleníti meg a találatokat

Kettőnél több programot is összeköthetünk csővezetékekkel.

A reguláris kifejezések

c Önmagát a *c* karaktert jelöli, amennyiben *c* nem speciális karakter.

\c A *c* karaktert jelenti attól függetlenül, hogy *c*-nek van-e speciális jelentése.

^ Sor elejére illeszt.

\$ Sor végére illeszt.

. Egy darab akármilyen karakter.

[abc] Az *a*, *b* vagy *c* egyike.

[^abc] Bármely karakter, amely nem *a*, *b* vagy *c*.

*r** Az *r* elem nullszor vagy sokszor ismétlődhet.

\(r) *r* reguláris kifejezés önmaga, de erre később hivatkozni lehet a *\n* alakban, ahol *n* számjegy.

\n Hivatkozás az *n*. *\(r)* alakú reguláris kifejezésre, ahol *n* egy számjegy. (Az első ilyenre a

\1 hivatkozik.)

r\{m\} *r* elem pontosan *m*-szer ismétlődhet.

r\{m,\} *r* elem legalább *m*-szer ismétlődhet.

tac – egy állomány sorait írja ki rendre, hátulról előre

tee – a standard bemenetén kapott adatokat a standard kimenetre és valamennyi argumentumként kapott fájlba másolja

grep – a paraméterként megadott fájlokra vagy a standard inputra illeszt reguláris kifejezéseket

`grep mit_keresünk hol_keresünk`

`grep -i` – nem különbözteti meg a kis- és nagybetűket

`grep -w` – a keresési minta önálló szó (előtte, utána szóköz van), vagyis ragozatlan alakot keres

`grep -n` – kiírja, hogy hányadik sorban található a keresési minta

`grep -v` – negálja a keresési feltételt

cut – oszlopokat vág ki
 cut -d*karakter* – szeparátorkarakter megadása
 cut -f – oszlop megadása [szám]

sort – szövegfájl sorainak rendezése
 sort -u – kiszűri az ismétlődéseket
 sort -r – csökkenő sorrendbe rendez
 sort -n – numerikus érték szerint rendez
 sort -k*mező* – adott oszlop szerint rendez
 sort -t*karakter* – szeparátorkarakter megadása

uniq – egy rendezett fájlból kiszedi a duplikált sorokat (csak rendezett állomány esetén működik!)

tr – karakterek lecserélése, tömörítése és/vagy törlése
 tr mit mire
 tr -d *karakter* – eltávolítja az adott karaktert

head – fájl első részének kiírása
 head -c 50 *fájlnev* – az első 50 karaktert írja ki
 head -2 *fájlnev* – első két sort írja ki

tail – kiírja a meghatározott fájl utolsó részét
 tail -15 *fájlnev* – utolsó 15 sort írja ki

wc – fájlokban található byte-ok, szavak és sorok számát írja ki
 wc -l *fájlnev* – sorok számát írja ki
 wc -w *fájlnev* – szavak számát írja ki

sed – folyamszerkesztő (stream **editor**): a megnevezett fájlokat (alapértelmezés szerint a standard bemenetet) a szabványos kimenetre másolja, de közben egy parancsokat tartalmazó szkriptnek megfelelően megszerkeszti
 sed *parancs input_fájl* – csak egy parancs esetén
 sed -e *parancs₁ -e ... parancs_n input_fájl* – több parancs esetén
 sed -f *parancsfájl input_fájl* – parancsfájlban összegyűjtött parancsok végrehajtása
 A két leggyakoribb parancs a csere (s) és a törlés (d).
 pl.: sed -e 's/nem/Nem/' *fájl*
 sed -e '/nem/d' *fájl*

Egyéb

wget – állományok letöltése
date – kiírja vagy beállítja a rendszerdátumot és időt
time – egy program futási idejét méri
who – megmutatja, ki van bejelentkezve
whoami – az aktuális felhasználó nevét írja ki
which – parancs helyének lekérdezése
find – fájlokat keres egy könyvtárstruktúrában
more – szűrőprogram fájl tartalmának oldalankénti kiírásához
less – hasonlít a more-ra, de visszafelé való mozgást is engedélyes, nem csak előremenőt
ping – tesztkérelést küld egy hosztnak
clear – a terminál képernyőtörlése (Ctrl+L)
su – indít egy shellt adminisztrátori módban (kilépésig érvényben marad!)
sudo – a parancs végrehajtásának idejére lép adminisztrátori módba (ideiglenes!)

A shell programozás alapjai

Szkripteket UNIX-on az úgynevezett **Shebang** sor vezeti be, mely a #! karakterek után a szkript értelmezőjének elérési útvonalát adja meg, pl. /bin/sh, /bin/bash.

Az idézőjelek használatáról

- A "dupla" idézőjelek között a shell nem értelmezi speciálisan a *, ?, [,], {, }, ;, <, >, stb. karaktereket (nincs globbing), viszont pl. működik a változókra való hivatkozás és a külső parancs kimenetének behelyettesítése.
- Az 'aposztrófok' között a shell semmilyen speciális karaktert nem értelmez, még a dollárjelet sem, úgyhogy tetszőleges szöveget írhatunk aposztrófok közé betű szerint.
- Magát az aposztróft sehogyan sem tudjuk aposztrófok közé írni, mivel a backslash is elveszíti a szokásos escape-funkcióját.
- A kétféle idézőjelen belül mindig szabadon használhatjuk a másik fajtát.
- A `backtick`ek közé írt parancssor standard outputját a shell behelyettesíti oda, ahol a backtickes kifejezés szerepelt, de jobb helyett a \$(zárójeles) alakot használni, mivel az szabadon egymásba ágyazható.

A változók használata

Alapvetően \$változónév, de ha pl. sztringet akarunk fűzni a változó tartalmához, akkor a hozzáfűzött sztring összefolyna a változónévvel. Ennek elkerülésére: \${változónév}.

A többi elvileg már nem kell!