

Héjprogramok  
Linux alatti parancsértelmezők  
1.0

Fehér János

2002. június 20.

Copyright © 2001-2002 Linux-felhasználók Magyarországi Egyesülete

E közlemény felhatalmazást ad önnek jelen dokumentum sokszorosítására, terjesztésére és/vagy módosítására a Szabad Szoftver Alapítvány által kiadott GNU Szabad Dokumentációs Licenz 1.1-es, vagy bármely azt követő verziójának feltételei alapján. Nem Változtatható Szakaszok nincsenek, Címlap-szövegek nincsenek, a Hátlap-szövegek neve pedig „hátlapszöveg”. E licenz egy példányát a GNU Szabad Dokumentációs Licenz elnevezésű szakasz alatt találja.

A módosított változat közzétételéért felelős személyek:

*Sári Gábor* saga@lme.linux.hu

Javítások: Sári Gábor

## **Szerző**

*Fehér János*

## **Szakmai lektor**

## **Nyelvi ellenőrzés**

*Sári Gábor* saga@tux.hu

## **Formázás (L<sup>A</sup>T<sub>E</sub>X)**

*Sári Gábor* saga@tux.hu  
*Kósa Attila* atkosa@shinwa.hu

**Előzmények**  
**Héjprogramok**  
**Linux alatti parancsértelmezők**

A kiadás éve: 2002.

**Szerző**

*Fehér János*

**Szakmai lektor**

**Nyelvi ellenőrzés**

*Sári Gábor* [saga@tux.hu](mailto:saga@tux.hu)

**Formázás (L<sup>A</sup>T<sub>E</sub>X)**

*Sári Gábor* [saga@tux.hu](mailto:saga@tux.hu)  
*Kósa Attila* [atkosa@shinwa.hu](mailto:atkosa@shinwa.hu)

# Tartalomjegyzék

<b>1. Héjprogramok - Linux alatti parancsértelmezők</b>	<b>4</b>
1.1. Bevezető . . . . .	4
1.2. A shell röviden . . . . .	4
1.3. A héjprogramok múltja, jelene, jövője . . . . .	5
1.3.1. A múlt . . . . .	5
1.3.2. A jelen . . . . .	5
1.3.3. A jövő . . . . .	6
1.4. A shellekről általában . . . . .	7
1.4.1. A shell működése . . . . .	7
1.4.2. A shell megváltoztatása . . . . .	8
1.4.3. Shell változók, környezeti változók . . . . .	9
1.5. A shellekről részletesen . . . . .	10
1.5.1. Bourne Shell . . . . .	10
Programindítás . . . . .	10
fájlelek . . . . .	11
A Bourne Shell és a változók . . . . .	12
1.5.2. Korn Shell . . . . .	13
Shell kapcsolók . . . . .	13
Helyettesítő nevek (alias) . . . . .	14
Parancsnapló (history) . . . . .	15
A Korn shell által módosított változók . . . . .	17
A Korn Shell változói . . . . .	17
1.5.3. Bourne Again Shell . . . . .	17

## 1. fejezet

# Héjprogramok - Linux alatti parancsértelmezők

### 1.1. Bevezető

A sorozat ezen füzetében az Olvasó betekintést nyerhet a Linux alatti héjprogramok használatába és programozásukba. Munkánkban nagy mértékben támaszkodunk a sorozat korábbi tagjaira, ezért teljesen kezdő felhasználóknak javasoljuk azok elolvasását, mert alapozó ismeretek nélkül sok felesleges és időrabló nehézségbe ütközhetnek.

Ahhoz, hogy kiadványunkat sikerrel forgassa, szükséges a megszerzett ismeretek gyakorlatban való kipróbálása is. Ehhez mindössze egy olyan számítógépre (vagy ilyen gépre szóló azonosítóra) lesz szüksége, amelyen az általunk taglalt héjprogramok megtalálhatók, vagy feltelepíthetők. Ehhez igyekszünk segítségükre lenni lemez mellékletünkkel, amelyen az általunk taglalt szoftvereknek mind a forrása, mind pedig a bináris változata megtalálható.

### 1.2. A shell röviden

Mi az a shell?

Az operációs rendszernek kiadott parancsokat beolvasó és értelmező programokat a Unix-terminológia shelleknek (héjprogram, burok) nevezi. Ezek alapvető jellegzetességei minden Unix-típusú rendszernek. Ennek oka a rendszer felépítésében keresendő, amely nagy vonalakban a rendszermagból és a köréje csoportosuló alkalmazásokból áll. Mint ismeretes, a kernel önmagában nem elég ahhoz, hogy dolgozni tudjunk, a külvilággal való interaktivitást a programoknak kell megvalósítaniuk. Ilyen kapcsolat-teremtő eszközök többek közt a shellek is.

A shell angolul kagylóhéjat, burkot jelent. Annak igazi okát, hogy annak idején miért éppen erre a szóra esett a Unix-készítők választása, mára homály fedi. Funkcionalitását tekintve a névválasztás igen találó:

- egy héj elrejt valamit a kíváncsi szemek elől, csökkentve ezzel a lehetőségét a nemkívánatos behatásoknak (védelem a felhasználtól)

- vannak díszesebb-szolidabb héjak, ám felépítésében egyik sem bonyolult (nem elrettentés a cél, hanem a könnyű kezelhetőség)
- egy burok adott célközönség számára készül (mint egy jó munkaeszköz)
- egy tengeri kagylót fülünkhöz emelve halljuk a tenger zúgását (kommunikáció a felhasználóval)

Számos egyéb értelmezés is elképzelhető, ezeket az Olvasó fantáziájára bízunk.

## 1.3. A héjprogramok múltja, jelene, jövője

### 1.3.1. A múlt

A legrégebbi, ám még ma is használt shell az ún. Bourne Shell, mely nevét atyjáról, az AT&T Bell Laboratories egykori munkatársáról, Steven Bourne-ről kapta. Az utóbbi harminc évben kevés kozmetikázáson átesett program kivétel nélkül minden Unix-típusú rendszeren fellelhető, amely egyik nagy előnye is a többivel szemben. „Tudása” szerénynek mondható, akárcsak erőforrás-igénye. A POSIX 1003.2-es és 1003.2a shell-specifikációk elkészítéséhez a Bourne Shell szolgált alapul.

Amikor a University of California at Berkeley is bekapcsolódott a Unix fejlesztésébe, új színfoltot hoztak a shellek világába. Megalkották a saját, Bourne Shellhez képest számos kényelmi funkcióval és C-nyelvvvel hasonlatos szintaktikájú parancsértelmezőjüket, a C Shellt (Egy anekdota szerint a név a „sea shell” (tengeri kagyló) szójátékból származik.). A fejlesztést Bill Joy végezte. A C Shell fejlesztése a '70-es évek végén új erőre kapott, ennek eredménye lett a TENEX C Shell, amelyet főképp Ken Greernek és Paul Place Waynek köszönhetünk.

A '80-as évek derekán David Korn vezetésével a Bell Laboratories továbblépett a Bourne Shellen, megalkotva a Korn Shellt. Ez volt a Unix System V ágának első modern shellje. Ennek készült egy nagyközönség számára is ingyenesen használható változata, a Public Domain Korn Shell. Mivel füzetünk elsősorban a Linuxot használókhhoz szól, ezért ez utóbbit fogjuk részletesen taglalni, ám az esetleges eltérésekre is felhívjuk a figyelmet.

A Free Software Foundation (Szabad Szoftver Alapítvány) megalakulása, a GNU-licenzelés új héjprogramot hívott életre, a Bourne Again Shellt. Mint neve is mutatja, ez a Bourne Shell GPL-licenzelésű, továbbfejlesztett változata, amelyet ma már az első számú linuxos shellként tarthatunk számon.

### 1.3.2. A jelen

Napjainkban számos egyéb shell is elérhető. Ezen új shellek közös jellemzője, hogy speciális körülményekhez tervezték őket. Jelentőségük nem számottevő, de egy rövid bemutatást mindegyik megér:

**Easy Shell** : egy kis méretű parancsértelmező, érdekessége a Lisp-szerű szintaktikája.

**Pdmenu** : tulajdonképpen nem is shellé állunk szemben, hanem egy programozható menüvel, amelyet - kényelmi szempontokat tekintve - olyan felhasználóknak célszerű beállítani, akiknek nem céljuk az alapvető ismeretek elsajátítása sem. Ismeri a színeket és az almenüket, amelyeket akár egérrel is kiválaszthatunk.

**RC :** ez az AT&T Plan 9 operációs rendszer shelljének egyik implementációja. Jellegzetessége a kis méret és gyorsaság, a C-nyelvű szintaktika (amely a C Shellnél sokkal jobban megközelíti a C-nyelvet) és egy hatékony mechanizmus a változók manipulálására.

**ASH :** a NetBSD Bourne Shellje, amely a POSIX shell-specifikációját is betartja, emellett olyan lehetőségeket is tartogat, mint az Emacs vagy vi-szerű parancsorszerkesztés, továbbá korábban beírt parancsainkat újra előhívhatjuk, valamint jobbjainkat is vezérelhetjük. Aki csak ezt a néhány plusz tulajdonságát használja a Bourne Again Shellnek vagy a többi általános shellnek, annak tudjuk ajánlani az ASH alapértelmezettkénti beállítását, mivel sokkal kevesebb függvénykönyvtárra támaszkodik, mint nagyobb tudású társaik és ez kevesebb erőforrásigényt is eredményez.

**Karel's Interactive Simple Shell :** szintén egy kis méretű interpreter. Különösen alkalmas biztonsági lemezek shelljeként, mivel olyan beépített funkciókkal rendelkezik, mint az alias, cat, cd, chgrp, chmod, chown, cp, exec, exit, grep, help, history, kill, ln, ls, mkdir, mknod, more, mv, printenv, pwd, read, rm, rmdir, setenv, sleep, source, touch, umount, unsetenv, ver, wc, where. Egyszóval minden olyan dolog megtalálható benne, amire egy biztonsági helyreállításnál szükségünk lehet.

**Stand-alone Shell :** akinek a Kiss nem tetszett, annak alternatíva lehet a Stand-alone Shell. Beépített parancsaival (-ar, -chattr, -chgrp, -chmod, -chown, -cmp, -cp, -dd, -echo, -ed, -grep, -file, -find, -gunzip, -gzip, -kill, -ln, -ls, -lsattr, -mkdir, -mknod, -more, -mount, -mv, -printenv, -pwd, -rm, -rmdir, -sum, -sync, -tar, -touch, -umount, -where) ugyanis igen hasznosnak bizonyulhat egy esetleges rendszerösszeomlás utáni helyreállításnál.

**Limited Shell :** ez egy kezdőknek szánt shell, DOS-kompatibilis parancsokkal. Többek közt beállítható a kis-nagybetű érzéketlenség a belső parancsokra, az egyes könyvtárak megadhatók úgy, hogy a DOS-os egybetűs meghajtónevekre is hallgassanak. Derült hangulatot okoz a globális konfigurációs állomány is, amely a keresztségben a /etc/autoexec nevet kapta :-)

**Operator's Shell :** vannak helyzetek, amikor arra van szükségünk, hogy a felhasználók mozgásterét lekorlátozzuk néhány parancsra. Ilyenkor jön jól ez a shell, amelynek konfigurációs fájljában megadhatók azok a programok, amelyeket a felhasználók elindíthatnak.

### 1.3.3. A jövő

A '90-es évek PC-ire a számítási kapacitás megnövekedtével gombamód szaporodtak a különböző grafikus felhasználói felületek (GUI-k). Pozitív tulajdonságaikkal szemben teljesen nyilvánvalóak hátrányaik is, úgy mint a felesleges erőforráshasználat, és egy idő után a lassú kezelhetőség. Ez utóbbihoz egy kevés magyarázat: a tapasztalat azt mutatja, hogy a shellekkel kapcsolatos gyakorlati tudás gyarapodásával sokkal gyorsabban érhetjük el célunkat, mint egérkattintással. Ennek oka az, hogy a tapasztaltabbak tíz ujjal gépelnek, és aktívan alkalmazzák a shellek kényelmi funkcióit (névkiegészítés, álnevek). Ezen érvek miatt a füzet szerzője nem hisz a shellek „kihalásának” elméletében, és szívből reméli, hogy az utolsó oldalak olvasásakor ezzel a Tisztelt Olvasó is egyetért majd.

## 1.4. A shellekről általában

### 1.4.1. A shell működése

Ahhoz, hogy a későbbi félreértéseket elkerüljük, pontosítani kell néhány fogalmat. Mivel az informatika nyelve angol, ezért a későbbi könnyebb tájékozódás kedvéért az egyes szakszavak angol megfelelője is megemlítsre kerül.

Az operációs rendszer működése közben a kommunikáció valamilyen terminálon (esetleg virtuális, emulált) keresztül történik. Ennek egyik eszköze lehet a shell is, melyen keresztül a vezérlés szöveges karakterekre korlátozódik. A vezérlő jelsoraink bevitelének elsődleges eszköze a terminál billentyűzete. Minden egyes billentyű lenyomásakor annak kódját a shell megkapja és egy pufferben tárolja, amelynek tartalmát többnyire szerkeszteni is lehet. A parancssor ezen szolgáltatását sorszerkesztőnek nevezzük. A sorszerkesztővel többnyire egyszerű műveleteket - törlést, felülírást, beszűrást - végezhetünk a kurzormozgató billentyűk és -kombinációk segítségével. Ezt követően a bevitt jelsor megjelenik a terminálon (echo). Egy érvényesítő végjellel (alapértelmezésként ez az ENTER vagy RETURN billentyű) lehet a parancsot értelmeztetni. Ilyenkor a puffer tartalmát megkapja a shell értelmező (interpreter) része, amely szétbontja utasításokra és paraméterekre. Az utasítások között elsőbbségek, precedenciák lehetnek, azaz nem feltétlenül beviteli sorrendben lesznek elindítva. Végrehajtás során, ha szintaktikailag helyes az utasítássor, akkor a shell először helyettesítő névként (alias), ha nincs ilyen, akkor a beépített parancskészletében (builtin) keresi meg a hozzátartozó „cselekvéssort”. Ha ilyen nem létezik, akkor egy alkalmazás nevéként értelmezi, és megkísérli elindítani. A beépített parancs a shellbe be van drótozva; működésében olyan, mintha egy programfájl kerülne végrehajtásra, azzal a különbséggel, hogy nem különálló részként található meg a fájlrendszeren. Egyes shellekben lehetőség nyílik a korábban kiadott parancsok tárolására, újbóli kiadására. Ezeket a parancsokat összefoglaló néven előzményeknek (history) nevezzük, az őket tartalmazó fájl pedig az előzményfájl.

Többnyire elmondható, hogy a végrehajtás szekvenciális, azaz egy újabb utasítás csak akkor adható ki, ha már egy korábbi lefutott. Ez igen nagy megkötés, ennek áthidalására szolgál a munkavezérlés (job control), amellyel megtehetjük, hogy bizonyos parancsokat háttérben futtassunk külön folyamatként, vagy felfüggesztjük. A felfüggesztett folyamatoktól a rendszer elveszi a CPU-t, ha szükséges, a lefoglalt memóriájukat elsők között írja ki a csere (swap) partícióra.

Hogy mit jelent az, hogy egy program előtérben vagy háttérben fut, nehéz elmagyarázni, ellenben szemléltetni jól lehet különbözőségekkel:

- Előtérben:
  - írhatja és olvashatja a terminált
  - a shell megvárja a jelenleg futó folyamat befejeztét, mielőtt új parancsbeviteli lehetőséget kínálna fel
- Háttérben:
  - nem képes olvasni a terminálról
  - írhat a terminálra, ha ezt stty to stop -pal le nem tiltjuk
  - nem vár az aktuálisan futó folyamat befejeztére, rögtön új parancsbeviteli módot kínál



Mindezekből leszűrhető, hogy a háttérben való futtatást olyan folyamatokhoz talál-  
ták ki, amelyek nem interaktívak, a terminált nem vagy csak alig használják. Példával  
élve, kedvenc szövegszerkesztőnket nem érdemes háttérben futtatni, mert folyamatos  
bevitelt vár, viszont egy kernelfordítást vagy tömörítést már célszerű.

Minden shell egy alkalmazás, amelyhez létezik egy fő futtatható állomány. Ezt  
elindítva egy külön folyamatként kel életre és teszi a dolgát addig, amíg ki nem lépnek  
belőle. Minden bejelentkezéskor egy ilyen programfájl hajtódik végre, melynek szülő  
folyamata az init, és minden, a shellből közvetlenül elindított többi folyamat őse a shell  
lesz.

### 1.4.2. A shell megváltoztatása

Mint már említettük, a shell egy program, tehát létezik legalább egy futtatható fájl,  
amivel el lehet indítani (Hogy melyek ezek, azt majd az egyes shellek leírásánál tár-  
gyaljuk.). Ezt kétféleképpen is megtehetjük: alapértelmezettként belépéskor a login  
program indítja el, vagy pedig a parancssorba begépeljük az fájl nevét. Attól füg-  
gően, hogy hogyan indult el, működése is eltérő lehet. A login program által indí-  
tottak (továbbiakban login shell) legtöbbször speciális konfigurációs fájlokat töltenek  
be, környezeti változókat vesznek át/állítanak be (lásd később), de ezen kívül minden  
egyébben megegyeznek a nem login shellekkel.

Előfordulhat, hogy a felhasználónk létrehozásakor megadott alapértelmezett  
(login) héjprogramunkat lecserélnénk egy másikra. Ehhez használjuk a `chsh` paran-  
csot. Debian 2.2 Linux alatt (a jelszavunk megadása után) ezt kiadva a következőt  
látjuk:

#### 1. példa : Aktuális héjprogramunk megváltoztatása

```
aries@taplo:~$ chsh
Password:
Changing the login shell for aries
Enter the new value, or press return for the default
Login Shell [/bin/bash]: /usr/bin/tcsh
```

Itt kell megadnunk az új shellünk futtatható állományát a teljes elérési úttal. Fon-  
tos, hogy ez többet megadjuk, hiszen csak a shell elindítása után lép érvénybe a  
PATH környezeti változó. Ha ezzel készen vagyunk, akkor a parancs módosítja a  
`/etc/passwd` (de más rendszereken ez eltérő lehet, például FreeBSD alatt ez a  
`/etc/master.passwd`) fájl a felhasználóra vonatkozó bejegyzését. Ez legtöbbször  
egyéb információkat is hordoz (User ID, Group ID, `chfn` paranccsal megadható  
egyéb adatok, stb.). Megeshet, hogy nem a sajátunkat akarjuk megváltoztatni, ekkor  
a `chsh` felhasználónév szintaxis használatos, például: `chsh manoe`. Mivel a  
parancs a jelszófájlt módosítja, ezért a változtatások csak újbóli bejelentkezés esetén  
lép életbe. A legtöbb terjesztésnél, ha nem létező, érvénytelen vagy egyéb okok mi-  
att nem futtatható programot adunk meg alapértelmezett shellünknek, akkor a rendszer  
felkínál egy lehetőséget, hogy egy másikat futtassunk (többnyire a Bourne Shellt adja  
be), ám igyekezzünk helyesen megadni új shellünket, hiszen ez nem mindenhol érvé-  
nyes! Csúnya meglepetésben lehet részünk régi Solarisokon, ha rossz adatokat adunk  
meg, főleg ha az illető login nevét éppen `root`-nak hívják...

Számos rendszerben létezik egy szövegfájl, amely az elérhető shell-ek futtatható  
állományainak listáját tartalmazza. Ez a `/etc/shells` állomány.

### 1.4.3. Shell változók, környezeti változók

Mivel a Unix-típusú rendszerek több felhasználóra és több program egyidejű használatára lettek kifejlesztve, ezért a shellekre különböző, „környezetből” jövő tényezők is hatnak. Ezek egy része általános, minden program által elérhető, változó értékű adatok, amelyeket globális vagy környezeti változóknak (environment variables) hívunk. Feladatuk az, hogy a shellben futó programok között egyfajta kommunikációs lehetőséget valósítsanak meg, mivel a shell gyermek folyamatai elérhetik az ő (shell) környezeti változóit. Általánosan elfogadott, hogy a környezeti változók nevei csak nagybetűket tartalmaznak. Léteznek dedikált környezeti változók, melyeknek jól meghatározott szerepük és értékük van. Ezek:

**EDITOR** Az alapértelmezett szövegszerkesztőnk futtatható állományának neve (elérési úttal, ha nincs benne a PATH-ben). Célszerű megadni kedvenc szövegszerkesztőnk, mivel számos program szerkesztési műveleteknél az itt megadott programot indítja el.

**HOME** A felhasználói könyvtárunk elérési útvonala. Ennek rövidítése a „/” jel.

**MAIL** Annak a mappának a neve (elérési útvonallal és fájlnevvvel), amelybe emailjeink érkeznek.

**OSTYPE** Az operációs rendszerünk típusa. Az adott rendszerre jellemző szöveget tartalmaz, pl. linux-gnu, FreeBSD, stb.

**PAGER** Az alapértelmezett fájlnéző programunk futtatható állományának nevét kell tartalmaznia. Sok program használja megjelenítéshez, többek közt a man is.

**PATH** Azokat az elérési útvonalakat tartalmazza, ahol a shell futtatható állományokat keres. Ez gyakorlatilag annyit tesz, hogy ahhoz, hogy ilyen könyvtárban lévő fájlokat futtatni tudjunk, nem szükséges azok elérési útvonalát megadni. Több könyvtárat úgy lehet megadni, hogy közéjük kettőspontot (: ) teszünk, tehát a szintaxis: PATH="könyvtár1:könyvtár2:...:könyvtárn". Például:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/bin/X11"
```

Fontos, hogy az éppen aktuális könyvtárat lehetőleg (ez fokozottan igaz azokra a felhasználókra, akik adminisztrátori jogosultságokkal bírnak) ne tegyük a PATH-be, mert ezzel biztonsági lyukat hozunk létre. Képzeljünk el egy olyan helyzetet, hogy rendszergazdaként az egyik felhasználó segítséget kér tőlünk, és mi készségesen próbálunk neki segíteni adminisztrátori azonosítókkal. Belépünk a felhasználó könyvtárába, beírjuk, hogy ls. Ám a felhasználónk gonosz módon ugyanitt elhelyezett egy futtatható ls programot, amely nem csak azt csinálja, hogy kiírja a fájlok listáját, hanem egyéb műveleteket is, pl. egy új, 0-ás UID-ű felhasználót is bejegyez a jelszófájlba, vagy a jelenlegi login programot felülírja egy olyanal, amely nem csak beléptet, hanem az azonosítót és a jelszót eljuttatja neki. Ekkor - ha az aktuális könyvtár benne van a PATH-ben - (emlékeztetőül, a Unix-típusú rendszerek „.” (pöttyel) jelölik az aktuális könyvtárat) nem a /bin-ben lévő ls hajtódik végre, hanem felhasználóé.

**PWD** Ez egy speciális változó, értékét a shellek minden könyvtárváltásnál frissítik és az aktuális könyvtár nevét tartalmazza a teljes elérési úttal. Ez legtöbbször megegyezik a pwd parancs visszatérési értékével, kivéve ha szimbolikus kötésű

(symbolic link) könyvtárról van szó. Ekkor ugyanis a PWD értéke abból az irányból fogja mutatni a könyvtárt, ahonnan beléptünk, míg a pwd parancs a fizikailag létező felől.

2. példa : Különbség a pwd parancs és a PWD környezeti változó között

```
aries@taplo:/usr/doc$ ls -l xbill
lrwxrwxrwx 1 root root 18 Jul 25 16:06 xbill -> ../share/doc/xbill

aries@taplo:/usr/share/doc$ cd /usr/doc/xbill
aries@taplo:/usr/doc/xbill$ pwd
/usr/share/doc/xbill
aries@taplo:/usr/doc/xbill$ echo $PWD
/usr/doc/xbill
```

Természetesen ez a különbség shellfüggő.

**SHELL** Jelenleg futó shellünk programneve teljes elérési útvonallal.

**TERM** a terminál típusát tartalmazza. Ez Linux esetén legtöbbször linux vagy xterm, de rendszertől függően ez lehet cons25, vt100, stb.

A shell karbantart egy belső paraméterlistát, amelyet shell változóknak hívunk. Ezeket más programok (szülő és gyermek egyaránt) nem érhetik el, nem módosíthatják és csak a shell működését befolyásolják. Általános megállapodás, hogy a shell változók nevei csak kisbetűkből állnak. Mivel ezek elég specifikusak, ezért róluk bővebben az egyes héjprogramok leírásánál értekezünk.

## 1.5. A shellekről részletesen

### 1.5.1. Bourne Shell

Mint azt a történelmi áttekintésben is elmondtuk, a Bourne Shell a legősibb shell. Bár az idők folyamán némi változtatáson ment keresztül, a felhasználók számára megmaradt annak, ami volt: egy egyszerű héjprogram, amely mentes a mai modern shellek minden kényelmi funkciójától. Nem véletlen azonban, hogy a mai napig megtalálható, hiszen felesleges fajsúlyosabb parancsértelmezőt elindítani, ha csak programokat akarunk elindítani vagy egy kisebb shell-szkripteket írni.

Ez a shell egy futtatható bináris fájlal rendelkezik, melynek neve: sh, amely a /bin könyvtárban található (ez is jól szemlélteti, hogy alapvető kelléke a Unix-típusú rendszereknek).

#### Programindítás

A shell működése fejezetben már tárgyaltuk, hogy elméletben hogyan is indul el egy program. Ez a gyakorlatban úgy működik, hogy a parancssorba beírjuk elérési úttal a futtatható állomány nevét, majd az érvényesítő karaktert (amelyet alapesetben az ENTER vagy a RETURN billentyű produkál). Ha a programunk futása véget ér vagy megszakad, újból megkapjuk a parancssort. Egy sorban több futtatandó fájlt is megadhatunk, ezek közé ; (pontosvessző) kell kerülni, tehát

```
program1 ; program2 [ ; program3 ... ]
```

Mint azt a shell működése részben már megemlítettük, létezik egy olyan futtatási lehetőség is, amikor a programot nem előtérben, hanem a háttérben futtatjuk. Ezt úgy tudjuk megtenni, ha a programnév után egy & (és) jelet teszünk:

```
program1 & [ program2 & ... ]
```

Hogy milyen folyamatok futnak a háttérben, azt a `jobs` paranccsal tudjuk lekérni. Ha önmagában adjuk ki a parancsot, akkor általában a

```
[szám1] szám2 visszatérési érték   kiadott parancs
```

formában egy listát kapunk, ahol a `szám1` a háttérben futó parancsunk sorszáma, a `szám2` pedig a folyamatazonosítója. A visszatérési érték a folyamat visszatérési értéke, amennyiben befelyeződött. Ha ez sikeresen történt, akkor többnyire az „Exit 0” szöveget láthatjuk. Megeshet, hogy a folyamat valamilyen jelzés miatt áll le, ekkor a jelzésre jellemző üzenetet kapunk, pl. Terminated, Killed stb.

3. példa :

```
aries@taplo:~/konyv$ find / -name 'dallas.mpeg' &
25723
aries@taplo:~/konyv$ killall -9 find
[1] 25723 Killed          find / -name dallas.mpeg
```

## Írásjelek

Minden héjprogram számára léteznek különleges jelentéssel bíró karakterek. Ezek legtöbbször különféle írásjelek, mint az egy- és kétszeres idézőjel, dollárjel rep (angolul backslash; a szerzőnek rendkívül tetszik ez a szellemes fordítás, melynek használatával is szeretné elismerését kifejezni a kortárs magyar nyelvújítóknak), stb.

- Escape-karakter

Az idézőjelek (lásd lentebb) nélküli rep (\) a Bourne Shell escape-karaktere. Többek közt ezzel a jellel lehet elérni, hogy az őt követő speciális karakter elveszítse módosító hatását. Ha a rákövetkező karakternek eddig sem volt különleges jelentése, akkor az nem jelenik meg. Ez alól kivétel az újsor karakter, amely szóközzé alakul át.

- Dollárjel

A dollárjellel (\$) tudunk különbséget tenni változó neve és annak értéke közt. Ha a változó neve előtt \$ áll, akkor annak értékére hivatkozunk.

- Idézőjelek

Az idézőjelek arra valók, hogy egyes karakterek vagy lefoglalt szavak speciális jellegét feloldják.

Az egyszeres idézőjelek (') közé zárt jelsoron belül semmilyen speciális karakter nem érvényesül. Egyszeres idézőjelek közé még egy csak úgy kerülhet, hogy egy reppel feloldjuk annak különleges jelentését.

A kétszeres idézőjelek (") közötti füzérben a dollárjelen, jobbratartó felsővesszőn (^) és repen kívüli különleges jelek elveszítik speciális jelentésüket. A \$ és a ` mindig, míg a rep csak abban az esetben lesz speciális, ha a „\$, „`”, „ ”, „\” vagy egy új sor követi.

A jobbratartó felsővessző tulajdonképpen egy operátor, amely a két ` (AltGr+7) között lévő jelsort végrehajtja, és a kimenetét adja vissza úgy, hogy az újsor karakter helyére szóközt rak.

- Kettőskereszt

A kettőskereszt (#) segítségével tudunk megjegyzéseket elhelyezni. Onnantól, hogy a # megjelenik egy sorban, az őt követő karaktereket az sh figyelmen kívül hagyja.

## A Bourne Shell és a változók

A Bourne Shell esetében nem különíthetők el teljesen a shell és a környezeti változók. Amikor a shell elindul, a környezeti változók táblából kiolvassa az információkat, majd definiál magának ugyanolyan nevű shell változókat (a konvenció miatt nagybetűseket) és bemásolja a hozzájuk tartozó adatokat. Tehát a shell csak a saját változóit használja, így ha egy shell változót módosítunk, akkor célszerű „kiexportálnunk” ugyanilyen nevű környezeti változóként is, hogy ne legyen keveredés, valamint, hogy a shellből indított programok is érzékeljék a változtatást. (Emlékeztető: a környezeti változókat a gyermek folyamatok is el tudják érni, a shell változókat nem.)

Változót létrehozni a értékadó utasítással lehet. Ennek szintaktikája:

```
valtozonév=érték
```

A változónév csak alfanumerikus karaktereket tartalmazhat, speciálisakat nem. Egy shell változót környezeti változóvá tenni az export belső paranccsal lehet:

```
export változónév
```

Ha egy változó értékére kívánunk a későbbiekben hivatkozni, akkor a változónév elé \$ (dollár) jelet kell tenni. Ha már nincs szükségünk a változóra, akkor az unset paranccsal szüntethetjük meg. Használata:

```
unset változónév
```

### 4. példa : Példa a Bourne Shell változókezelésére I.

```
aries@dynamo> valtozo=ertek
aries@dynamo> export valtozo
aries@dynamo> echo $valtozo
ertek
```

Megtehetjük, hogy több változónak vagy skalárnak az értékét akarjuk egyetlen sorban összefűzni. Ekkor el kell magyarázni a shellnek, hogy hol van vége a változónévnek, illetve, hogy hol kezdődik egy új vagy a skalár érték:

```
parancs skalár1${változónév1}${változónév2}
skalár2
```

Ha jól elhatárolható (speciális karakter), hogy hol van vége a változóknak, akkor a kapcsos zárójeleket elhagyhatjuk.

#### 5. példa : Példa a Bourne Shell változókezelésére II.

```
aries@taplo:~/~/$ alma="alma" ; bor="bor"
aries@taplo:~/~/$ mitiszunk=$alma$bor
aries@taplo:~/~/$ echo $mitiszunk
almabor
aries@taplo:~/~/$ miteszunk=$almapaprika
aries@taplo:~/~/$ echo $miteszunk
almapaprika

aries@taplo:~/~/$ miteszunk=${alma}paprika
aries@taplo:~/~/$ echo $miteszunk
almapaprika

aries@taplo:~/~/$ echo `ls -al /lost+found/`
total 20
drwxr-xr-x  2 root root 16384 Jun 24 17:05 .
drwxr-xr-x 26 root root  4096 Jul 24 10:10 ..
```

### 1.5.2. Korn Shell

A Korn Shell a Bourne Shell továbbfejlesztése. Rengeteg kényelmi funkciót építettek bele, amelyre már nagy igény volt (többek közt a Korn Shell képes arra, hogy „egy lépésben” hozzon létre környezeti változót.

#### 6. példa : Értékkadás és exportálás egy utasítással

```
export ALMA="jonatan"
```

A Korn Shell futtatandó állományának neve ksh.

#### Shell kapcsolók

A shell kapcsolók használata egy újabb mód arra, hogy megváltoztassuk héjprogramunk működését. Ezek voltaképpen kétállású kapcsolók: vagy érvényben vannak, vagy nem. Egy kapcsolót be- és kikapcsolni a

```
set -o opció"
```

formában lehet. Hogy melyik mire hivatott, azt az alábbi lista tartalmazza. Amelyik mellett parancssori kapcsoló található, azt a ksh után megadva a kívánt hatás szintén elérhető.

**bg** A háttérbe kerülő folyamatok alacsonyabb prioritással futnak. Ez általában alapértelmezett, de hogy mekkora az új prioritás, az terjesztéstől és operációs rendszertől is függ.

**emacs** EMACS módba kapcsolja a sorszerkesztőt. Hatására a parancsnaplóban és a parancssorban alkalmazhatjuk az EMACS szövegszerkesztő funkcióit.

**gmacs** Gosling EMACS módba kapcsolja a sorszerkesztőt. Ez többé-kevésbé megegyezik az EMACS móddal.

**vi** Visual Editor (vi) módba kapcsolja a sorszerkesztőt. Bekapcsolásával a vi funkciói használhatók. Egy parancs addig szerkeszthető, amíg a 033-as kódú escape karakter bevitelre nem kerül.

**ignoreeof** Aktív állapotban nem engedi, hogy fájlvége (EOF - End Of File) karakterre kilépjünk a shellből. Az exit vagy a logout parancsokat kell ilyenkor alkalmazni.

**markdirs** Fájlnévkiegészítés során ha könyvtárról van szó, a kiegészített név végére tesz egy perjelet (/).

**noclobber -C** Megakadályozza a kimenet átirányítását már létező fájlba.

**monitor -m** Engedélyezi a folyamatvezérlést. Azok a többfeladatos operációs rendszerek és interaktív shellek esetében ez általában alapértelmezett.

**errexit** Nullától eltérő hibakód esetén a kivétel „elkapása”. Ez shellszkriptek esetén annak befejeződését vonja maga után.

**keyword -k** Bekapcsolt állapotban ha egy parancs paramétere a név=érték formájú, akkor a shell levágja azt a parancssorból, és végrehajtja azt, mint a parancsot megelőző utasítást, majd a parancs számára a végrehajtás ideje alatt kiexportálja.

**nolog** Hatására a függvénydefiníciókat nem menti el a parancsnapló fájlba.

**noglob** Aktív állapotban a fájlnevek nem kerülnek kiértékelésre.

**noexec -n** Hatására a parancsok beolvasást követően szintaxisellenőrzésen esnek át, de nem kerülnek végrehajtásra.

**nounset -u** Shellprogramozás esetén hasznos hibakereső lehetőség, amely révén figyelmeztetést kapunk, ha olyan változóra hivatkozunk, amely nem hordoz értéket (nincs definiálva).

**trackall -t** Ha a kapcsoló aktív, akkor minden érvényes, nem teljes bejárési úttal megadott parancshoz automatikusan hozzárendelődik egy nyomkövető álnév (lásd ott).

### Helyettesítő nevek (alias)

A helyettesítő nevekkel parancsok valódi neve és azok paraméterei helyettesíthetők, így egy hosszabb, sűrűn használt parancsot nevezhetünk el egy rövidebb névvel.

Használata:

```
alias [-t] [-x] név="parancs [paraméterek]"
```

7. példa : Helyettesítő név

```
$ alias ls="ls -al"
$ ls
total 4
drwxr-xr-x  2 aries  wheel   512 Jul 31 10:32 .
drwxr-xr-x  4 aries  wheel   512 Jul 31 10:32 ..
```

**-t** A kapcsoló segítségével „nyomkövető álneveket” (tracked aliases) definiálhatunk. A shell számos kiadott parancshoz automatikusan készít egy nyomkövető álnév bejegyzést. Ezek értéke megegyezik az őket definiáló parancshoz tartozó bejárési úttal és a fájlnévvel. Mivel a shellek előbb nézik végig az álneveket, mint a PATH környezeti változóban megadott bejárési utakat, ezért ennek teljesítménynövelő hatása van. Ha nem adunk meg helyettesítendő nevet, akkor egy listát

kapunk:

#### 8. példa : Nyomkövető álnevek

```
$ alias -t
cat=/usr/bin/cat
ls=/usr/bin/ls
vi=/usr/bin/vi
```

```
$ alias -t man
cat=/usr/bin/cat
ls=/usr/bin/ls
man=/usr/bin/man
vi=/usr/bin/vi
```

**-x** Ezzel a kapcsolóval az export paranccsal kiexportált álneveket definiálhatunk. Ezeket a shell gyermek folyamatai is megkapják, azokat használhatják.

Ha már nincs szükségünk egy helyettesítő névre, akkor az unalias paranccsal szüntethetjük meg.

```
unalias helyettesítő név
```

vagy

```
unalias -a
```

A -a kapcsoló hatására minden helyettesítő név törlődni fog. Mivel a helyettesítő nevek a memóriában vannak tárolva, ezért a sűrűn használt álneveket célszerű a belépéskor lefutó `.profile`-ban elhelyezni.

#### Parancsnapló (history)

Ismer parancsnapló lehetőséget, amely beépített funkcióként elérhető a history paranccsal, így megtehetjük, hogy egy korábbi parancsot megváltoztatunk és kiadunk anélkül, hogy újra be kellene gépelnünk. Ez úgy lehetséges, hogy a shell folyamatosan menti minden kiadott parancsunkat egy ún. history fájlba, melynek helyét és nevét a HISTFILE változó hordozza. Ennek a fájlnak a mérete véges, szabályozni a HISTSIZE változóval lehet, amelynek értéke a letárolandó programok száma.

A kiadott utasításainkkal kapcsolatos műveletekre a history parancsot kell használni. Ez egy helyettesítő név az `fc` (ez a „fix command”, magyarul „parancs javítás” rövidítése) parancsra a más Bourne-ági shellek érdekében. A teljesség kedvéért mindkettőt ismertetjük.

```
history [-r] [innentől [idáig]]
```

vagy

```
fc -l [-nr] [innentől [idáig]]
```

Kapcsolók:

**-l** Hatására az `fc` kilistázza a parancsnapló tartalmát. A history az `fc -l` álneve.



**-n** Ennek eredményére az `fc` nem tünteti fel a parancsok melletti sorszámot. Ez hasznos lehet, ha pl. fájlba irányítjuk át a listát, hogy újra kiadjuk.

**-r** A kapcsoló eredményéül a listánk fordított sorrendben jelenik meg (a legutóbb kiadott kerül legfelülre).

**innentől, idáig** Ezek megadásával szabályozható, hogy a parancsnapló mekkora része jelenjen meg.

A parancssorba paraméterek nélkül beírva kapunk egy listát a korábban bevitt parancsokról.

9. példa : History parancs paraméterek nélkül

```
$ history
123 cd globt2
124 ls
125 ls -l
126 df
127 quit
128 w
129 df
130 man ksh
131 less
132 man ksh
133 vim
134 joe
135 ls /usr/bin
136 ls /usr/local/bin
137 vi
138 history
```

Ha csak egy bizonyos részt szeretnénk megkapni, akkor meg kell adni, hogy mettől meddig terjedjen a listázás:

10. példa : Parancsnapló egy részének lekérése

```
$ history 123 128
123 cd globt2
124 ls
125 ls -l
126 df
127 quit
128 w
```

A már kiadott parancsainkat újra felhasználni az `r` paranccsal lehet.

```
r [régi=új] [parancs]
```

vagy

```
fc -e - [régi=új] [parancs]
```

**parancs** Ha tudjuk a parancsunk sorszámát, akkor azt parancsunk megadva újra le fog futni. Ha negatív számot adunk meg, akkor az aktuális sorszámhoz hozzáadva az új sorszámú fog megjelenni.

**régi, új** Ha ezt a két kapcsolót megadjuk, akkor amennyiben a régi jelsort megtalálja a megadott parancsban, kicseréli az újra. Ez hasznos lehet, ha elgépeztünk egy hosszú parancsot, és tudjuk, hogy mit kell rajta módosítani.

## 11. példa : Régebben kiadott parancs módosított újrakiadása

```
Legyen a parancsunk a 125-ös sorszámú ls -l.  
Ezt szeretnénk kicserélni ls -al -re.  
Ekkor nem kell mást tennünk, mint kiadni az  
r -l=-al 125 utasítást.
```

### A Korn shell által módosított változók

A Korn Shell rendelkezik néhány speciális változóval, amivel a Bourne Shell nem. Ezek értéke folyamatosan változik, és ő maga frissíti. Ezek:

**ERRNO** Azt a nemnulla számot tartalmazza, amellyel a shellben legutoljára futtatott program visszatért. A változó hasonlít a \$?-hez, azzal a különbséggel, hogy csak akkor kap értéket, ha a program hibával tér vissza.

**LINENO** Csak szkriptfuttatásnál van jelentősége, ugyanis annak a sornak a számát veszi fel, ahol a végrehajtás tart.

**OLDPWD** A legutolsó cd parancs előtt könyvtárnév teljes elérési útvonalát tartalmazza.

**PPID** A jelenlegi folyamat szülőjének folyamatazonosítója. Ha a \$\$ az éppen aktuális folyamat PID-jét tartalmazza, akkor a PPID a folyamat szülőjének PID-jét. Főleg szkriptek írásánál használatos.

**RANDOM** Értéke 0 és 32767 közti szám. Minden lekéréskor véletlenszerűen változik az értéke. Mivel a véletlenszám generálása nem függ a bemenettől, ezért shellszkriptekben is jól használható.

**SECONDS** A shell indulásától számított időt adja vissza másodpercekben.

### A Korn Shell változói

A Korn Shell változói magában foglalják a Bourne Shell-ét is, de tartalmaznak számos egyebet is.

### 1.5.3. Bourne Again Shell

A Bourne Again Shell (továbbiakban BASH) mint a

# GNU Szabad Dokumentációs Licenz 1.1 verzió, 2000 március

Copyright ©2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Jelen licenz szó szerinti sokszorosítása és terjesztése bárki számára megengedett, változtatni rajta ugyanakkor nem lehet.

## 0. ELŐSZÓ

Jelen Licenz célja egy olyan kézikönyv, tankönyv, vagy effajta írott dokumentum megalkotása, mely a szó szoros értelmében „szabad”: annak érdekében, hogy mindenkinek biztosítsa a szöveg sokszorosításának és terjesztésének teljes szabadságát, módosításokkal, vagy anélkül, akár kereskedelmi, akár nem-kereskedelmi úton. Másfelől, a Licenz megőrzi a szerző, vagy kiadó munkája elismeréséhez fűződő jogát, s egyúttal mentesíti őt a mások által beiktatott módosítások következményei alól.

Jelen Licenz egyfajta „etalonnak” tekinthető, ami nem jelent mást, mint hogy a dokumentumból származtatott munkák maguk is szabad minősítést kell, hogy kapjanak. E dokumentum egyben a GNU Általános Felhasználói Licenz kiegészítőjeként is szolgál, mely egy a szabad szoftverekre vonatkozó etalon licenz.

E Licenzet a szabad szoftverek kézikönyveiben való használatra alkottuk, hiszen a szabad szoftver egyben szabad dokumentációt is igényel: egy szabad programot olyan kézikönyvvel kell ellátni, mely ugyanazon szabadságokat biztosítja, mint maga a program. Jelen Licenz, mindazonáltal, nem korlátozódik pusztán kézikönyvekre; feltételei tetszőleges tárgykörű írott dokumentumra alkalmazhatók, függetlenül attól, hogy az könyvformában valaha megjelent-e. Mindamellet e Licenzet főként olyan munkákhoz ajánljuk, melyek elsődleges célja az útmutatás, vagy a tájékoztatás.

## 1. ALKALMAZHATÓSÁG ÉS DEFINÍCIÓK

E Licenz minden olyan kézikönyvre, vagy más jellegű munkára vonatkozik, melyen megtalálható a szerzői jogtulajdonos által feltüntetett figyelmeztetés, miszerint a dokumentum terjesztése jelen Licenz feltételei alapján lehetséges. A „Dokumentum” alább bármely ilyen jellegű kézikönyvre, vagy egyéb munkára vonatkozik. A lakosság minden tagja potenciális licenztulajdonosnak tekinthető, és mindegyikük megszólítása egyaránt „ön”.

A Dokumentum „Módosított Változata” bármely olyan munkára vonatkozik, mely tartalmazza a Dokumentumot, vagy annak elemeit akár szó szerint, akár módosításokkal, és/vagy más nyelvre lefordítva.

A „Másodlagos Szakasz” egy egyedi névvel bíró függelék, esetleg a Dokumentum egy megelőző szakasza, mely kizárólag a kiadóknak, vagy az alkotóknak a Dokumentum átfogó tárgyköréhez (vagy kapcsolódó témákhoz) fűződő viszonyáról szól, és nem tartalmaz semmi olyat, ami közvetlenül ezen átfogó témakör alá eshet. (Ha például a Dokumentum részben egy matematika tankönyv, úgy a Másodlagos Szakaszban nincs lehetőség matematikai tárgyú magyarázatokra.) A fenti kapcsolat tárgya lehet a témakörrel, vagy a kapcsolódó témákkal való történelmi viszony, illetve az azokra vonatkozó jogi, kereskedelmi, filozófiai, etikai, vagy politikai felfogás.

A „Nem Változtatható Szakaszok” olyan speciális Másodlagos Szakaszok számítanak, melyek illetően való meghatározását az a közlemény tartalmazza, miszerint a Dokumentum jelen Licenz hatálya alatt lett kiadva.

A „Borítószövegek” olyan rövid szövegrészek, melyek Címlap-szöveggé, illetve Hátlap-szöveggé kerülnek felsorolásra abban a közleményben, miszerint a Dokumentum jelen Licenz hatálya alatt lett kiadva.

A Dokumentum „Átlátszó” példánya olyan géppel-olvasható változatot jelöl, mely a nyilvánosság számára hozzáférhető formátumban kerül terjesztésre, továbbá melynek tartalma szokványos szövegszerkesztő-programokkal, illetve (pixelekből álló képek esetén) szokványos képmegjelenítő-programokkal, vagy (rajzok esetén) általánosan hozzáférhető rajprogramok segítségével azonnal és közvetlenül megtekinthető, vagy módosítható; továbbá olyan formátumban mely alkalmas a szövegszerkesztőkbe való bevitelre, vagy a szövegszerkesztők által kezelt formátumokba való automatikus átalakításra. Egy olyan, egyébként Átlátszó formátumban készült példány, melynek markujja úgy lett kialakítva, hogy megakadályozza, vagy eltántorítsa az olvasókat minden további módosítástól, nem tekinthető Átlátszónak. A nem „Átlátszó” példányok az „Átlátszatlan” megnevezést kapják.

Az Átlátszóság kritériumainak megfelelő formátumok között megtalálható például a markup nélküli egyszerű ASCII, a Texinfo beviteli formátum, a  $\LaTeX$  beviteli formátum, az SGML vagy az XML egy általánosan hozzáférhető DTD használatával, és a standardnak megfelelő, emberi módosításra tervezett egyszerű HTML. Az Átlátszatlan formátumok közé sorolható a PostScript, a PDF, a szabadalmaztatott és csak fizetős szövegszerkesztőkkel olvasható formátumok, az olyan SGML vagy XML, melyhez a szükséges DTD és/vagy egyéb feldolgozó eszközök nem általánosan hozzáférhetők, és az olyan gépileg-generált HTML formátum, melyet egyes szövegszerkesztők hoznak létre, kizárólag kiviteli célra.

Egy nyomtatott könyv esetében a „Címlap” magát a címlapot, illetve bármely azt kiegészítő további oldalt jelöl, amely a jelen Licenzben definiált címlap-tartalmak közzétételéhez szükséges. Az olyan formátumú munkáknál, melyek nem rendelkeznek effajta címlappal, a „Címlap” a munka címéhez legközelebb eső, ám a szöveg törzsét megelőző szövegrészeket jelöli.

## **2. SZÓ SZERINTI SOKSZOROSÍTÁS**

Önnek lehetősége van a dokumentum kereskedelmi, vagy nem-kereskedelmi jellegű sokszorosítására és terjesztésére, bármely médiumon keresztül, feltéve, hogy jelen Licenz, a szerzői jogi figyelmeztetés, továbbá a Dokumentumot jelen Licenz hatálya

alá rendelő közlemény minden példányban egyaránt megjelenik, és hogy e feltételeken kívül semmi mást nem tesz hozzá a szöveghez. Nem alkothat olyan technikai korlátokat, melyek megakadályozhatják, vagy szabályozhatják az ön által terjesztett példányok elolvasását, vagy sokszorosítását. Mindazonáltal elfogadhat bizonyos összeget a másolatok fejében. Amennyiben az ön által terjesztett példányok száma meghalad egy bizonyos mennyiséget, úgy a 3. szakasz feltételeinek is eleget kell tennie.

A fenti kritériumok alapján kölcsönbe adhat egyes példányokat, de akár nyilvánosan is közzéteheti a szöveget.

### **3. SOKSZOROSÍTÁS ÉS NAGYOBB MENNYISÉGBEN**

Amennyiben 100-nál több nyomtatott változatot tesz közzé a Dokumentumból, és annak License feltételül szabja a Borítószovegek meglétét, úgy minden egyes példányt köteles ellátni olyan borítólappal, melyeken a következő Borítószovegek tisztán és olvashatóan fel vannak tüntetve: Címlap-szövegek a címlapon, illetve Hátlap-szövegek a hátlapon. Mindkét borítólappal egyértelműen és olvashatóan rá kell vezetnie a kiadó, vagyis jelen esetben az ön nevét. A címlapon a Dokumentum teljes címének jól láthatóan, továbbá minden egyes szónak azonos szedésben kell megjelennie. Ezen felül, belátása szerint, további részleteket is hozzáadhat a borítólappalhoz. Amennyiben az esetleges módosítások kizárólag a borítólappalokat érintik, és feltéve, hogy a Dokumentum címe változatlan marad, továbbá a borítólappalok megfelelnek minden egyéb követelménynek, úgy a sokszorosítás ettől eltekintve szó szerinti reprodukciónak minősül.

Abban az esetben, ha a borítólappalok bármelyikén megkövetelt szövegrészek túl hosszúnak bizonyulnának az olvasható közzétételhez, úgy csak az elsőként felsoroltakat kell feltüntetnie (amennyi józan belátás szerint elfér) a tényleges borítón, a továbbiak pedig átkerülhetnek a következő oldalakra.

Amennyiben 100-nál több Átlátszatlan példányt tesz közzé, vagy terjeszt a Dokumentumból, úgy köteles vagy egy géppel-olvasható Átlátszó példányt mellékelni minden egyes Átlátszatlan példányhoz, vagy leírni minden egyes Átlátszatlan példányban egy a módosítatlan Átlátszó példányt tartalmazó nyilvános hozzáférésű számítógéphálózat elérhetőségét, ahonnan bárki, anonim módon, térítésmentesen letöltheti azt, egy közismert hálózati protokoll használatával. Ha az utóbbi lehetőséget választja, köteles gondoskodni arról, hogy attól a naptól kezdve, amikor az utolsó Átlátszatlan példány is terjesztésre került (akár közvetlenül ön által, akár kiskereskedelmi forgalomban), a fenti helyen közzétett Átlátszó példány még legalább egy évig hozzáférhető legyen a felhasználók számára.

Megkérjük, ámde nem kötelezzük önt arra, hogy minden esetben, amikor nagyobb példányszámú terjesztésbe kezd, már jóval ezt megelőzően lépjen kapcsolatba a Dokumentum szerzőivel, annak érdekében, hogy megkaphassa tőlük a Dokumentum esetleges felújított változatát.

### **4. MÓDOSÍTÁS**

Önnek lehetősége van a Dokumentum Módosított Változatának sokszorosítására és terjesztésére a 2. és 3. szakaszok fenti rendelkezései alapján, feltéve, hogy a Módosított

Változatot kizárólag jelen Licenz feltételeivel összhangban teszi közzé, ahol a Módosított Változat a Dokumentum szerepét tölti be, ezáltal lehetőséget biztosítva annak terjesztésére és módosítására bárkinek, aki csak hozzájut egy példányához. Mindezen felül, a Módosított Változat az alábbi követelményeknek is meg kell, hogy feleljen:

- A Címlapon (és ha van, a borítókon) tüntessen fel egy a Dokumentumétól, illetve bármely korábbi változattól eltérő címet (melyeknek, ha vannak, a Dokumentum Előzmények szakaszában kell szerepelniük). Egy korábbi változat címét csak akkor használhatja, ha annak szerzője engedélyezte azt.
- A Címlapon szerzőkként sorolja fel a Módosított Változatban elvégzett változtatásokért felelős személyeket, vagy entitásokat, továbbá a Dokumentum fő szerzői közül legkevesebb ötöt (vagy mindet, ha nincsenek öten).
- A Címlapon a Módosított Változat közzétételéért felelős személyt tüntesse fel kiadóként.
- A Dokumentum összes szerzői jogi figyelmeztetését hagyja érintetlenül.
- Saját módosításaira vonatkozóan is tegyen közzé egy szerzői jogi megjegyzést, a többi ilyen jellegű figyelmeztetés mellett.
- Rögtön a szerzői jogi figyelmeztetéseket követően tüntessen fel egy közleményt, az alábbi Függelék mintájára, melyben engedélyezi a Módosított Változat felhasználását jelen Licenz feltételei alapján.
- A fenti közleményben hagyja érintetlenül a Nem Változtatható Szakaszok és a szükséges Borítószövegek jelen Dokumentum licenzében előírt teljes listáját.
- Mellékelje jelen Licenz egy eredeti példányát.
- Az „Előzmények” szakaszt, illetve annak címét szintén hagyja érintetlenül, emellett adjon hozzá egy új elemet, amely minimálisan tartalmazza a Módosított Változat címét, kiadási évét, továbbá az új szerzők, illetve a kiadó nevét, a Címlapon láthatókhöz hasonlóan. Amennyiben a Dokumentum nem tartalmaz semmiféle „Előzmények” elnevezésű szakaszt, úgy hozzon létre egyet, mely tartalmazza a Dokumentum címét, kiadási évét, továbbá a szerzők, illetve a kiadó nevét, a Címlapon láthatókhöz hasonlóan; majd ezt követően adjon hozzá egy új, a Módosított Változatra vonatkozó elemet, a fentiekkel összhangban.
- Ne tegyen változtatásokat a Dokumentumban megadott Átlátszó példány nyilvános hálózati elérhetőségét (ha van ilyen) illetően, vagy hasonlóképp, a Dokumentum alapjául szolgáló korábbi változatok hálózati helyére vonatkozóan. Ezek az „Előzmények” szakaszban is szerepelhetnek. Csak abban az esetben hagyhatja el egyes korábbi változatok hálózati elérhetőségét, ha azok legkevesebb négy évvel a Dokumentum előtt készültek, vagy ha maga az alkotó engedélyezi azt.
- Bármely „Köszönetnyilvánítás”, vagy „Ajánlások” szakasz címét hagyja érintetlenül, továbbá gondoskodjon arról, hogy azok tartalma és hangvétele az egyes hozzájárulókat, és/vagy az ajánlásokat illetően változatlan maradjon.
- A Dokumentum összes Nem Változtatható Szakaszát hagyja érintetlenül, úgy címüket, mint tartalmukat illetően. A szakaszok számozása, vagy bármely azzal egyenértékű jelölés nem tartozik a szakaszcímek közé.

- Töröljön minden „Jóváhagyás” elnevezésű szakaszt. Effajta szakaszok nem képezhetik részét a Módosított Változatnak.
- Ne nevezzen át semmilyen létező szakaszt „Jóváhagyás”-ra, vagy olyasmire, mely címében a Nem Változtatható Szakaszokkal ütközhet.

Ha a Módosított Változat új megelőző szakaszokat tartalmaz, vagy olyan függelékkeket, melyek Másodlagos Szakaszok minősülnek, ám nem tartalmazzak a Dokumentumból származó anyagot, abban az esetben, belátása szerint, e szakaszok némelyikét, vagy akár az összeset nem változtathatóként sorolhatja be. Ehhez nem kell mást tennie, mint felsorolni a szóban forgó címeket a Módosított Változat licenzének Nem Változtatható Szakaszok listájában. E címeknek határozottan el kell különülnie minden egyéb szakaszcímtől.

„Jóváhagyás” elnevezésű szakaszt csak akkor adhat a Dokumentumhoz, ha az kizárólag a Módosított Változatra utaló megjegyzéseket tartalmaz – például mások re-  
 cenzióira vonatkozóan, vagy hogy egy szervezet a szöveget egy standard mérvadó definíciójaként ismerte el.

Címlap-szöveg gyanánt egy legfeljebb öt szóból álló szövegrészt adhat meg, a Hátlap-szöveg esetén pedig 25 szót fűzhet a Módosított Változat Borítószövegeinek végéhez. Bármely entitás csak és kizárólag egy Címlap- és egy Hátlap-szövegrészt adhat (akár közvetítőn keresztül) a Dokumentumhoz. Ha a dokumentum már eleve rendelkezik Borítószöveggel, akár azért, mert azt korábban ön adta hozzá, vagy mert valaki más önön keresztül gondoskodott erről, abban az esetben nincs lehetőség újabb Borítószöveg hozzáadására; a régit mindazonáltal lecserélheti, abban az esetben, ha annak kiadója egyértelműen engedélyezi azt.

A Dokumentum szerzője/i és kiadója/i jelen Licenz alapján nem teszik lehetővé nevük nyilvános felhasználását egyetlen Módosított Változat támogatása, vagy támogatottsága érdekében sem.

## 5. KOMBINÁLT DOKUMENTUMOK

Önnek lehetősége van a Dokumentum egyéb, e Licenz hatálya alatt kiadott dokumentumokkal való kombinálására a 4. szakasz módosított változatokra vonatkozó rendelkezései alapján, feltéve, hogy a kombináció módosítás nélkül tartalmazza az eredeti dokumentumok összes Nem Változtatható Szakaszát, és hogy azok mind Nem Változtatható Szakaszokként kerülnek felsorolásra a kombinált munka licenzében.

A kombinált munkának jelen Licenz mindössze egy példányát kell tartalmaznia, az egymással átfedésben lévő Nem Változtatható Szakaszok pedig kiválthatók egy összegzett példánnyal. Amennyiben több Nem Változtatható Szakasz szerepelne ugyanazon címmel, ám eltérő tartalommal, úgy alakítsa át minden egyes szakasz címét olyan módon, hogy mögéírja zárójelben az eredeti szerző és kiadó nevét (ha ismeri), vagy egy egyedi sorszámot. Ha szükséges, a Nem Változtatható Szakaszok címeivel is végezze el a fenti módosításokat a kombinált munka licenzében.

A kombinált munkában az eredeti dokumentumok összes „Előzmények” elnevezésű szakaszát össze kell olvasztania, miáltal egy összefüggő „Előzmények” szakasz jön létre; hasonlóképp kell eljárnia a „Köszönetnyilvánítás”, illetve az „Ajánlások” szakaszok tekintetében. Ugyanakkor minden „Jóváhagyás” elnevezésű szakaszt törölnie kell.

## 6. DOKUMENTUMGYŰJTEMÉNYEK

Önnek lehetősége van a Dokumentumból, illetve bármely egyéb, e Licenz hatálya alatt kiadott dokumentumból gyűjteményt létrehozni, és az egyes dokumentumokban található licenzeket egyetlen példánnyal kiváltani, feltéve, hogy a gyűjteményben szereplő összes dokumentum esetén minden más tekintetben követi jelen Licenz feltételeit, azok szó szerinti sokszorosítására vonatkozóan.

Tetszése szerint ki is emelhet egy meghatározott dokumentumot a gyűjteményből, továbbá terjesztheti azt jelen Licenz feltételei alapján, feltéve, hogy a szóban forgó dokumentumhoz mellékeli e Licenz egy példányát, és minden egyéb tekintetben betartja jelen Licenz előírásait a dokumentum szó szerinti sokszorosítására vonatkozóan.

## 7. ÖSSZEFŰZÉS FÜGGETLEN MUNKÁKKAL

A Dokumentum és annak származékainak különálló, vagy független dokumentumokkal, illetve munkákkal való összefűzése egy közös tárolási, vagy terjesztési egységen, egészében nem tekinthető a Dokumentum Módosított Változatának, feltéve, hogy az összefűzés nem lesz szerzői jogvédelem. Az effajta összefűzés eredményeként „összegzés” jön létre, ám jelen Licenz nem érvényes az abban a Dokumentummal együtt szereplő önálló munkákra, hacsak azok nem a Dokumentum származékai.

Amennyiben a 3. szakasz Borítószövegekre vonatkozó rendelkezései alkalmazhatók a Dokumentum e példányaira, és a Dokumentum a teljes összegzésnek kevesebb, mint egynegyedét teszi ki, úgy a Dokumentum Borítószövegeit olyan módon is el lehet helyezni, hogy azok csak magát a Dokumentumot fogják át. Minden más esetben a teljes összegzés borítólapjain kell feltüntetni a fenti szövegeket.

## 8. FORDÍTÁS

A fordítás egyfajta módosításnak tekinthető, így hát a Dokumentum lefordított példányai a 4. szakasz rendelkezései alapján terjeszthetők. A Nem Változtatható Szakaszok lefordítása külön engedélyt igényel a szerzői jogtulajdonostól, mindazonáltal közzéteheti a lefordított változatokat is abban az esetben, ha az eredeti Nem Változtatható Szakaszokat is belefoglalja a munkába. E Licenz lefordítására ugyanezek a feltételek érvényesek, vagyis a lefordított változat csak akkor jelenhet meg, ha mellette ott van az eredeti, angol nyelvű Licenz szövege is. Amennyiben eltérés mutatkozna az eredeti változat, illetve a fordítás között, úgy a Licenz angol nyelvű eredetije tekintendő mérvadónak.

## 9. MEGSZŰNÉS

A jelen Licenzben egyértelműen kijelölt kereteken kívül tilos a Dokumentum bármilyen sokszorosítása, módosítása, allicenzelése, vagy terjesztése. Minden ezzel szembeni sokszorosítási, módosítási, allicenzelési, vagy terjesztési kísérlet a jelen Licenzben meghatározott jogok automatikus megszűnését vonja maga után. Azok a fe-



lek, ugyanakkor, akik önön keresztül jutottak másolathoz, vagy jogosultságokhoz, nem veszítik el azokat, amíg maradéktalanul betartják e Licenz előírásait.

## **10. JELEN LICENZ JÖVŐBENI JAVÍTÁSAI**

Megtörténhet, hogy a Szabad Szoftver Alapítvány időről időre felülvizsgálta és/vagy új verziókat bocsát ki a GNU Szabad Dokumentációs Licenzből. E verziók szellemisége hasonló lesz jelen változathoz, ám részleteikben eltérhetnek, új problémák, új aggályok felmerülése okán. Vö.: <http://www.gnu.org/copyleft/>

A Licenz minden változata egyedi verziószámmal van ellátva. Ha a Dokumentum jelen Licenz egy konkrét, számozott verziójára, „vagy bármely újabb verzióra” hivatkozik, úgy önnek a szóban forgó változat, vagy bármely újabb a Szabad Szoftver Alapítvány által (nem vázlatként) publikált verzió feltételeinek követésére lehetősége van. Ha a Dokumentum nem ad meg semmilyen verziószámot, úgy bármely a Szabad Szoftver Alapítvány által valaha (nem vázlatként) publikált változat megfelel.

## **FÜGGELÉK: A Licenz alkalmazása saját dokumentumaira**

Ha e Licenzet egy ön által írt dokumentumban kívánja használni, akkor mellékelje hozzá a Licenz egy példányát, továbbá vezesse rá az alábbi szerzői jogi és licenz közleményeket, rögtön a címlapot követően:

Copyright © ÉV AZ ÖN NEVE.

E közlemény felhatalmazást ad önnek jelen dokumentum sokszorosítására, terjesztésére és/vagy módosítására a Szabad Szoftver Alapítvány által kiadott GNU Szabad Dokumentációs Licenz 1.1-es, vagy bármely azt követő verziójának feltételei alapján. A Nem Változtatható Szakaszok neve **SOROLJA FEL A CÍMLAP- SZÖVEGEK NEVE LISTA**, a Címlap-szövegek neve **LISTA**, a Hátlap-szövegek neve pedig **LISTA**. E licenz egy példányát a „GNU Szabad Dokumentációs Licenz” elnevezésű szakasz alatt találja.

Ha a szövegben nincsenek Nem Változtatható Szakaszok, úgy írjon „nincs Nem Változtatható Szakasz”-t, ahelyett, hogy egyenként felsorolná azokat. Ha nincsenek Címlap-szövegek, akkor írjon „nincs Címlap-szöveg”-et, ahelyett, hogy „a Címlap-szövegek neve LISTA”, és hasonlóképp járjon el a Hátlap-szövegek esetében is.

Amennyiben a dokumentum haladó programkód-példákat is tartalmaz, úgy azt javasoljuk, hogy e példákat egy választása szerinti szabad szoftver licenz alatt közölje – mint például a GNU Általános Felhasználói Licenz –, hogy lehetővé tegye a kódok szabad szoftverekben való alkalmazását.

# Hátlapszöveg

Ezen dokumentum eredetije készült 2001-2002-ben a *Linux-Felhasználók Magyarországi Egyesülete* gondozásában a *MEH IKB* pénzügyi támogatásával. A dokumentum szabadon terjeszthető és másolható a *GNU Szabad Dokumentációs Licenz* feltételei alapján.