

Számítógép Architektúrák

2. Gyakorlat

(shell és HTML alapok)

OS.

- **Operációs rendszernek** (rövidítése gyakran **OS** az angol *operating system* forma alapján) nevezzük a számítástechnikában a számítógépeknek azt az alapprogramját, mely közvetlenül kezeli a hardvert, és egy egységes környezetet biztosít a számítógépen futtatandó alkalmazásoknak (például szövegszerkesztők, játékok stb.).
- A kezelt hardvererőforrásoknak része többek között a **memória**, a **processzor**, a **merevlemez** és a **perifériális** eszközök használata.

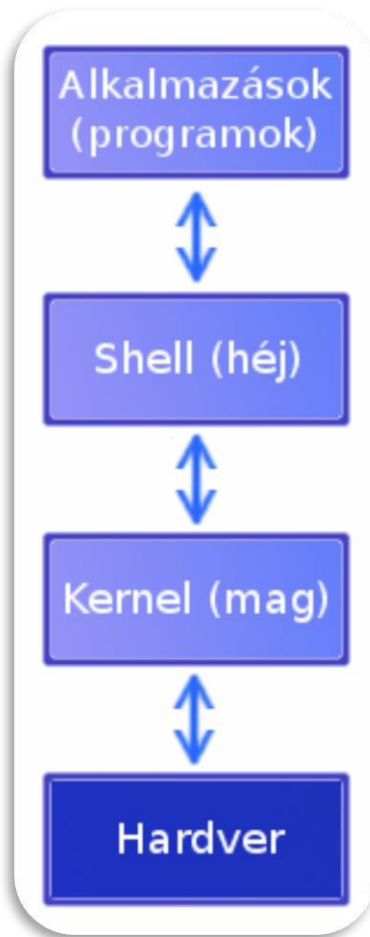
OS.

- Adalékként idézzük az ISO nemzetközi szabványosítási szervezet definícióját, mely szerint az **operációs rendszer**
- „Olyan programrendszer, amely a számítógépes rendszerben a programok végrehajtását vezérli: így például ütemezi a programok végrehajtását, elosztja az erőforrásokat, biztosítja a felhasználó és a számítógépes rendszer közötti kommunikációt.”
- *(fájlrendszer? Absztrakt szemléletmód? ...)*

OS. Felépítése

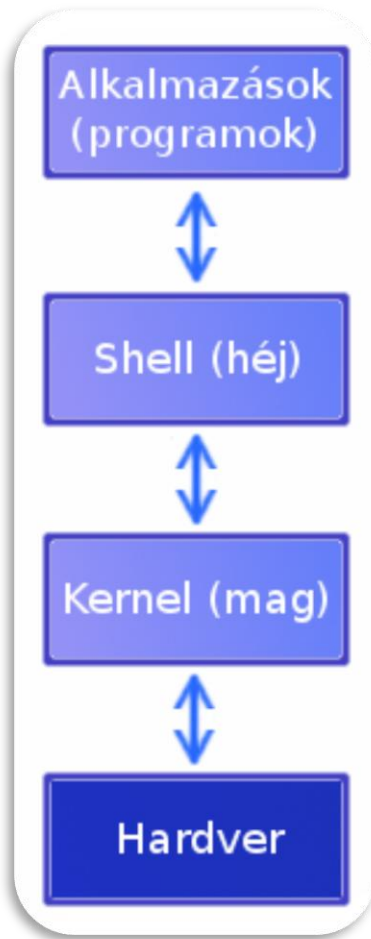
- Az operációs rendszerek alapvetően három részre bonthatók: a **felhasználói felület** (a *shell*, amely lehet egy grafikus felület, vagy egy szöveges), **alacsony szintű segédprogramok** és a **kernel** (mag) amely közvetlenül a hardverrel áll kapcsolatban.
- Vannak olyan operációs rendszerek, melyekben a *kernel* (mag) és a *shell* (héj) különválnak, így lehetőség nyílik ezek kombinációjára. Más rendszereknél, mint például a Windowsnál ez a különválasztás mind elméleti, mind gyakorlati szempontból nehezebb.
- Négyféle *kernel*típus ismert, az úgynevezett **monolitikus** kernel a **hibrid** kernel **mikrokernel**, és a kevésbé ismert **exokernel**.

A kernel feladatai:



- Ki- és bemeneti eszközök kezelése (billentyűzet, képernyő stb.)
- Programok, folyamatok futásának kezelése
 - Indítás, futási feltételek biztosítása, leállítás
 - Memória-hozzáférés biztosítása
 - Processzor idejének elosztása
 - Virtuális gép mutatása a nemkívánt taszkok felé (pontosabban a hardver által biztosított lehetőségek szoftverkiegészítései)
- Háttértárolók kezelése
- Rendszerhívások kiszolgálása
- Fájlrendszer (hierarchikus)
- egyéb

A *shell* feladatai:



- Kapcsolattartás a felhasználóval (felhasználói felület)
- Alkalmazások futásának kezelése (indítás, futási feltételek biztosítása, PIT leállítás)

Operációs rendszer indítása

- Az operációs rendszer feladatai közé tartozik, hogy felkészítse a számítógépet az alkalmazások futtatására. Az indítási folyamat során pontosan ez történik. Miután a számítógép öntesztje lefutott, a BIOS keresni kezdi az operációs rendszert a rendszertöltő szektorokban, és az elsőt, amit megtalál elindítja. Elsőként a kernel töltődik be és indul el. Innentől rendszerfüggő a folytatás, de nagy vonalakban általánosítható.
- **Eszközök inicializálása:** A rendszer keresi az új hardvereszközöket, a meglévőkhöz pedig betölti és elindítja az illesztő programot. (DOS-ban pl. config.sys)
- **Szolgáltatások elindítása**
- **Felhasználói interakció megkezdése**
 - **Bejelentkezés:** Több felhasználós rendszerek esetén mielőtt a felhasználó kapcsolatba lépne a számítógéppel, azonosítania kell magát, általában felhasználói névvel és jelszóval.
 - **Automatikusan induló programok indítása:** Ide tartoznak azok a programok, amik segítik a felhasználó kapcsolattartását a számítógéppel, illetve azok is, amiket ő maga jelölt ki. Tipikus vállalati példa a bejelentkezés után induló e-mail kliens.

Konzol

- „Majdnem minden számítógéphez tartozik egy billentyűzet és egy monitor, amely a számítógép része. A Unix szempontjából ez a billentyűzet és monitor együttes csak egy további terminál. Azonban egy speciális nevet adtak neki, ez a KONZOL. Bizonyos értelemben a konzol be van építve a számítógépbe, az összes többi terminál különálló, és csatlakoztatni kell a számítógéphez.
- Egy tipikus Unix rendszer a rendszergazda irodájában lévő hostot használhat. Ehhez a számítógéphez egy teremnyi terminál kapcsolódhat. A rendszergazda a konzolt használja - a beépített billentyűzetet és a monitort - a munkája során, mert ez a legkézenfekvőbb a számára. Mindenki más közönséges terminálokat használ.
- Azonban a Unix szempontjából a konzolnak semmilyen kitüntetett szerepe sincs. A rendszergazda éppúgy használhatna a munkájához egy terminált is.
- Felmerülhet a kérdés, hogy szükséges-e egyáltalán, hogy egy Unix rendszernek legyen konzolja? A válasz: egyáltalán nem. Néhány számítógéphez eredetileg nem is tartozik billentyűzet és monitor. Ekkor a rendszergazda másokhoz hasonlóan egy közönséges terminált használ, és a host számítógép, amely csak egy egyszerű doboz, lehet, hogy el van zárva valahol."

Konzol

- A személyi számítógépekben nincs beépített monitor és billentyűzet, csak lehet hozzájuk csatlakoztatni. A Linux használatához nem szükséges egyik sem. Itt sem játszik kiemelt szerepet a konzol. Ide érkeznek azok az üzenetek, amiket a rendszer küld a bootolás folyamán. A rendszerüzenetek is érkezhetnek a konzolra, ha így állítottuk be a rendszerünket. Létezik xkonzol is, ami a grafikus felületen jeleníti meg a konzolra küldött üzeneteket.
- A Linux lehetővé teszi, hogy a fizikailag egyetlen számítógépet virtuálisan úgy használjuk, mintha több különböző számítógépen (terminálon) dolgoznánk. Természetesen egyszerre csak egy terminálon tudunk dolgozni, de az elindított programok párhuzamosan futnak egymással. Az egyes virtuális konzolok között váltogathatunk a bal Alt-F? (?=1...12) billentyűkombinációval. Linux alatt 63 virtuális konzolt használhatunk egyszerre (alapértelmezetten 6 indul).
- A hálózaton vagy grafikus felületen keresztül bejelentkezett felhasználókhöz úgynevezett pseudo-terminálokat rendel a rendszer, ahol a billentyűzet és a képernyő annak a gépnek a billentyűzetéhez és képernyőjéhez rendelődik, ami előtt a felhasználó ül.

Unix rendszerháj (Shell)

- A Unix operációs rendszerben különféle parancssori értelmezők működnek, melyeket **rendszerhájnak** vagy angolul **shellnek** nevezünk. Ez a rendszerháj az összekötő kapocs a rendszer magját képező kernel és a felhasználó között. Az asztali rendszereken ezeket többnyire grafikus felületek egészítik ki.
- A shellből (konzolról) vezérelt gépek előnye, hogy jóval kisebb erőforrásigényűek a grafikus felületeknél. A grafikus felhasználói felületek (GUI) ugyanis egyrészt újabb réteget képeznek a rendszermag és a felhasználó között, másrészt erőforrásigényük többnyire magas. Ennek következtében a shellből (konzolról) vezérelt rendszerek stabilabbak, és gyorsabbak, így teherbírásuk miatt napjainkban főleg Unix alapú kiszolgálókon népszerűek.

Unix rendszerháj (Shell)

- Unix-típusú rendszerek alatt a felhasználó több rendszerháj között (**sh, bash, ksh, tcsh, csh ...**) is választhat, melyek mindegyike rendelkezik egyedi tulajdonságokkal, melyek segítségével bizonyos feladatok könnyebben elvégezhetőek, mint más rendszerhájakkal. A számítógépbe történő bejelentkezés után a rendszerháj indítja el az operációs rendszernek nélkülözhetetlen folyamatokat és automatikusan induló programokat.
- A shell programozható az úgynevezett **shellscriptek** segítségével. Ez egy speciális parancsnyelv, mely az operációs rendszer által értelmezhető utasításokat tartalmaz, így a klasszikus értelemben vett programozási nyelvektől eltérően nincs szükség a kód lefordítására és linkelésére.

A shell általános képességei

- saját parancsok kivitelezése
- „wildcard”-ok (speciális illesztések) használata fájlnevekben: *, ?
- héjprogramok ("szkript"-ek) végrehajtása
- feltétels kifejezések (*if, case*) és ciklusok (*while, for*) alkalmazása
- beépített parancsok használata (*cd, read*)
- pszeudo-parancsok létrehozása (*alias*)
- belső változók használata
- a környezet manipulálása új folyamatok indítása előtt
- bemenet és kimenet átirányítása
- több folyamat indítása, ezek adatcseréje csöveken (pipe) keresztül
- folyamatok indítása a háttérben
- a jelentkezési karakter definiálása (prompt)
- külső parancsok keresési sorrendjének megadása

A shell általános képességei

- Parancssor szerkesztés (*command line editing*) és kiegészítés (*completion*)
- korábbi parancsok visszakeresése és szerkesztése (*command history*)
- folyamatok megállítása és újraindítása (*Job Control*)
- számítási feladatok elvégzése (*expr*)
- állománytulajdonságok tesztelése (*test*)
- jelzések (signal) küldése (*kill*) és fogadása (*trap*)
- más folyamat bevárása (*wait*)

Parancsok

- *Fehér karakterekkel határolt* szavak sora.
- Első szó a *parancs* neve,. – többi szó az argumentumok.
tail -n+3 fájl
0 1 2
- Az sh (bash, ksh, tcsh ...) beolvassa, értelmezi, átalakítja, végrehajtja ...
- Külső v. belső parancsok listája?
- Indul új burok processz a parancsnak, vagy nem?

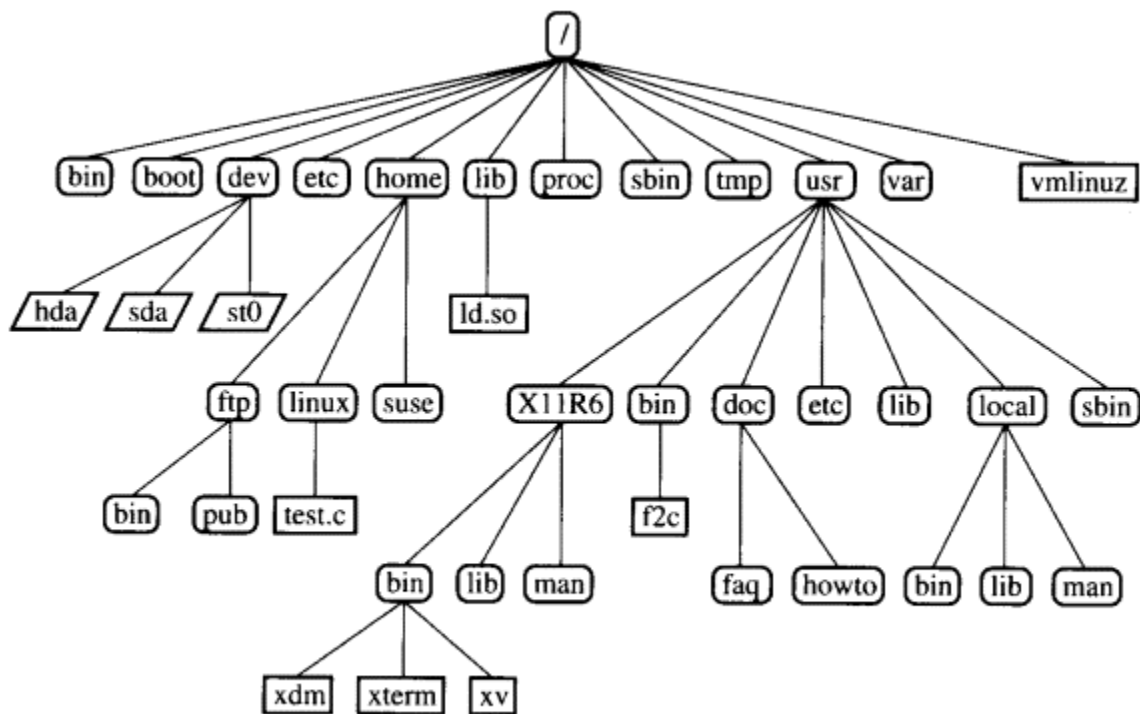
Pár „fontosabb” parancs (case sensitive!!!)

Parancs	Hatása
man	A parancs manual oldalait nyitja meg, rövid, tömör, célratörő leírás
man -k szó	Minden olyan parancsot megad, mely manual oldalában szerepel a "szó"
<i>parancs</i> --help	Információ a "parancs" használatáról
whatis <i>parancs</i>	Egysoros a parancsról.
whereis <i>parancs</i>	Hol is van a parancs?
<i>which</i> parancs	A program futtatható állományának elérési útvonalát adja meg (általában /usr/bin)
file:///usr/share/doc	böngészőbe írva, a legtöbb telepített program leírását megtalálhatjuk

Állománykezelés

Parancs	Hatása
ls	A könyvtárstruktúrát jeleníti meg.
cd	Könyvtár váltás parancs.
mkdir	Jegyzék létrehozás.
rmdir	Könyvtár törlés (ha üres)
cp -r /honnan/mit /hova/milyen_néven	Másolás
mv /honnan/mit /hova/milyen_néven	Fájlokat könyvtárakat mozgat, vagy nevez át
cat	Fájl tartalmát írja ki.
chmod	Fájlok-, könyvtárakra vonatkozó jogokat állíthatjuk be.
rm	Állományok eltávolítása.

Átlagos Linux fájlrendszer



Átlagos Linux fájlrendszer

- A UNIX, ill. Linux operációs rendszer könyvtárstruktúrája is olyan mint egy fa, ami fejjel lefelé terebélyesedik. Számos könyvtárból áll. Ezeknek a tartalma egyes UNIX verziókban, ill. Linux disztribúciókban egymástól némiképp eltér. Pl. a weblap oldalait tartalmazó HTML fájlokat a **Debian** disztribúció a **/var/www** könyvtárban, míg a **SuSE** összeállítása a **/usr/local/httpd/htdocs** könyvtárban tárolja.
- A felhasználó szempontjából **három kitüntetett könyvtár** bír elsődleges fontossággal:
 - **root** directory
 - **home** directory
 - **current** directory

/	A főkönyvtár, a könyvtárfa kiindulópontja
/home	Ehhez csatlakoznak a felhasználók könyvtárai
/dev	A hardvereszközöket reprezentáló fájlok
/etc	A legfontosabb konfigurációs fájlok
/usr/bin	Közhasználatú parancsok
/sbin	A superuser és a rendszerindító parancsok
/usr/doc	Dokumentációs fájlok
/usr/local/man	Kézikönyv oldalak
/tmp	Ideiglenes fájlok
/var	Konfigurációs fájlok (Linkek az /usr-ből)
/lib	Megosztott programkönyvtárak
/proc	A processzek fájlrendszere

Ami még biztos kell majd...

Parancs	Hatása
more	Lapszűrő. Oldalanként tördel.
cut	Bement (stdin), vagy paraméterként megadott fájl minden sorának egy megadott részét vágja ki
diff -u file1 file2 > eredmény	Összehasonlítja a fájlok tartalmát, a különbséget pedig az eredmény-be írja
du	Az aktuális könyvtár fájljainak méretét adja meg
df	Szabad terület számítása, partícióként
echo szoveg	Kiírja a képernyőre a szoveg-et
find	Keresés
file sajt	megvizsgálja a sajt fájl típusát
grep	Szövegrészleteket keres fájlokban, valamint a kimenetben.
head	Szűrő eszköz. A fájl első 10 sorát írja ki.
tail	Szűrő eszköz. A fájl utolsó sorait írja ki.
tail -n+3 fájl head -n1 head -n3 fájl tail -n1	Egy fájl harmadik sorát így írhatjuk ki

Ami még biztos kell majd...

Parancs	Hatása
mc	Midnight Commander fájlkezelő
mcedit	Az mc szövegszerkesztője
ln -s	soft link-et hoz létre
mount	Eszköz felcsatlakoztatás (CD,partíció, pendrive, hdd, etc.)
umount <i>/dev/eszköz</i>	Lecsatlakoztatás
pwd	Kiírja az aktuális könyvtárat
sed	Stream editor, folyamatszerkesztő. A bemenetet a kimenetre másolja miközben megszerkeszti.
sync	A Ramból a merevlemezre még ki nem írt adatok szinkronizálása (Fontos)
touch <i>fájlnév</i>	létrehoz egy üres állományt
tree	Könyvtárstruktúrát írja ki
wc <i>fájl</i>	sor, szó, karakter számítása

Ami még biztos kell majd...

Parancs	Hatása
quota	Tároló kvótázás programja
who	Bejelentkezett felhasználók kiírása
whoami	Milyen néven is jelentkeztünk be?
users	Kiírja az rendszeren levő felhasználók nevét
finger	Felhasználó információi
talk	Felhasználókkal való kommunikálás egyik eszköze
last	Ki jelentkezett be utoljára és honnan (felhasználó vagy terminál alapján)
passwd	A bejelentkezett felhasználó jelszavának módosítása
parancs & fg : bg :	felélesztés, majd előtérben futás felélesztés, majd háttérben futás
kill " <i>pid</i> "	Folyamat leállítása PID szerint
time -v parancs	A parancs lefutási idejét méri

Linux fájljogosultság

- A Linux fájlrendszer tárolja a fájl tulajdonosának azonosítóját a fájlhoz tartozó csoportokat és a hozzáférési jogosultságot is. A hozzáférési jogosultságok ábrázolásához egy 3 jegyű számból álló kódot használ, amit fájlmodnak nevezünk.
- Első szám a saját (**u**ser) jogot
- Második szám a csoport (**g**roup) jogot
- Harmadik a többiek (**o**thers) jogait rögzíti
- (Nulladik helyen esetenként a speciális jogok állnak, erről később)

Linux fájljogosultság

- A saját jog alatt a fájl tulajdonosának jogait értjük, legtöbbször ő az adott fájl vagy könyvtár létrehozója is. Minden komponens a következő részekből áll 3x(read,write,execute):
 - Saját r w x | 4 2 1 [r=4; w=2; x=1 ; rwx=7 ...]
 - Csoport r w x | 4 2 1
 - Többiek r w x | 4 2 1

Linux fájljogosultság

- **r** (Read): Olvasási jog (vagyis az adott fájl ezáltal olvasható)
- **w** (Write): Írási jog (az adott fájl ezáltal válik írhatóvá)
- **x** (Executable): Végrehajtási jog (Futási)

Linux fájljogosultság

- **Fájlok** esetében a végrehajtási jognak csak a **futtatható fájloknál** van jelentőségük (bináris állományok, scriptek).
- **Könyvtárak** esetén az olvasási jog azt jelenti, hogy elolvashatja a fájlok neveit az adott könyvtárban, az írási jog jelenti, hogy a könyvtárban állományt, könyvtárat hozhatunk létre, míg a **futtatási jog megengedi a belépést a könyvtárba.**

Jogosultságok megváltoztatása

- Egy fájlt tulajdonosi (hozzáférési) jogait csak a fájl tulajdonosa, vagy a rendszergazda tudja megváltoztatni.

chmod +|-<mód> <fájlnév>

- **chmod 655 ~/public_html/index.html**
(rwx magunknak, rx csoportnak, rx másoknak az index.html-re)
- chmodnál meg kell határozni az alábbiakat:
Adunk vagy elveszünk jogot
(+ : adunk, - : elveszünk)
- Kinek/kitől (saját, csoport, mások (ugo))
Milyen jogot adunk (r w x / 4 2 1)

- **u** – saját (User)
chmod u+w munka.tar.gz Saját magunknak írási jog
- **g** – csoport (Group)
chmod o+x munka.tar.gz Másoknak futtatási jog
- **o** – mások (Other)
chmod o+x,u+w munka.tar.gz
- **a** - mindenki (All)
chmod 777 munka.tar.gz Mindenkinek minden jog, ugyanaz mint:
chmod a+rwx <fájlnév>
chmod 700 munka.tar.gz Csak nekem van jogom mindenhez, ugyanaz mint
chmod u+rwx,g-rwx,o-rwx <fájlnév>

Linux fájljogosultság

- Amikor egy fájlt létrehozunk, akkor az a jogosultságoknak egy alapértelmezett csoportjával fog rendelkezni. Pl.: Létrehozunk egy üres fájlt:

-

```
#touch akarmi  
#ls -la akarmi
```

rw-r--r--

-	rwX	rw-	r--
Fájltípus	Tulajdonos jogai	Csoport jogai	Mások jogai

ssh

- A **Secure Shell** (röviden: **SSH**) egy szabványcsalád, és egyben egy protokoll is, amit egy helyi és egy távoli számítógép közötti biztonságos csatorna kiépítésére fejlesztettek ki. Nyilvános kulcsú titkosítást használ a távoli számítógép hitelesítésére, és opcionálisan a távoli számítógép is hitelesítheti a felhasználót.
- Az SSH-t leggyakrabban arra használják, hogy egy távoli gépre belépjenek vele és parancsokat adjanak ki, de támogatja a tunnelinget , azaz tetszőleges TCP portok és X11 kapcsolatok továbbítását; fájlok biztonságos átvitelére is használható a kapcsolódó SFTP (Secure FTP) és SCP (Secure Copy) protokollok segítségével. Az SSH szerverek alapértelmezésben a 22-es TCP portot figyelik.

ssh

- Linux alatt szervere: **openssh-server**
- Linux kliense: **ssh**

- Windows alatt szervere: **freeSSHD**
- Windows alatt kliense: **Putty**

ssh

- **ssh** *kliens.cime*
> ssh jerry.iit.uni-miskolc.hu
- **ssh** *user@kliens.cime*
> ssh broda@nec13.iit.uni-miskolc.hu
- *Még nagyon sok lehetőség rejlik az ssh-ban.
X11 átirányítás, port átirányítás...*