Operációs rendszerek

Informatika elméleti alapjai

Horváth Árpád

2014. október 17.

1. A számítógépes rendszer

1.1. A hardver és a szoftver

A számítógépes rendszer szintjei - HW

- felhasználói programok
- rendszerprogramok
- hardver (hardware, HW)
  - gépi nyelv
    - 50–300 utasítás: adatmozgatás, aritmetikai (+,*) és összehasonlító műveletek ...
  - fizikai eszközök

A szoftverek (SW) csoportosítása

- felhasználói program: közvetlenül a felhasználó által megoldani kívánt feladatot old meg
- rendszerprogram: a számítógép működését szervezi
  - felhasználói módú rendszerprogramok
    - pl. parancsértelmező (shell), ablakkezelő rendszer*, fordítók, szövegszerkesztő (text editor)
  - *operációs rendszer* (Operating System, OS)
    - erőforrásokat kezel
    - alapokat biztosítja a felhasználói programok írásához

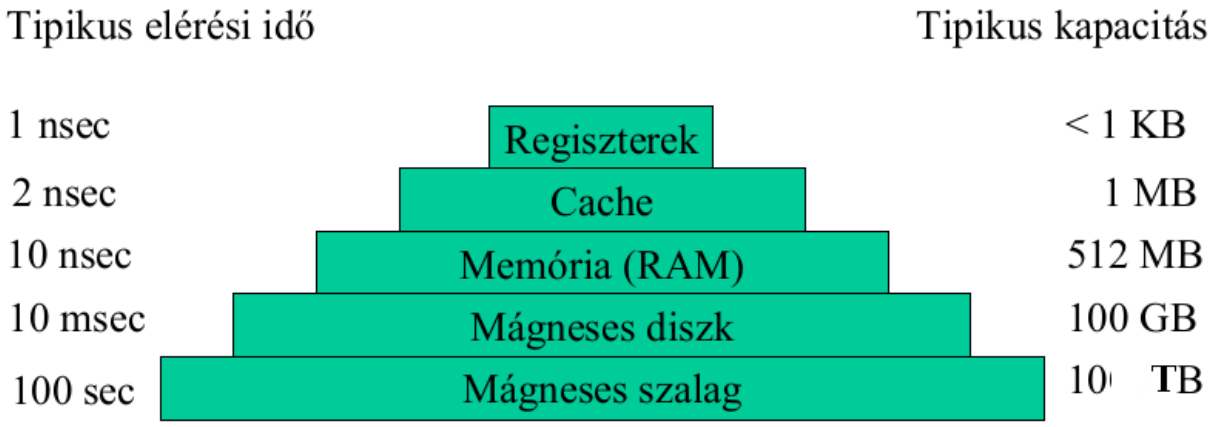
* Az ablakkezelő rendszerek egyes operációs rendszerek részét alkotják, más esetekben (például a Linux esetén) az operációs rendszer felett helyezkednek el. Linux esetén – például szerverekre – gyakran nem is telepítünk grafikus felületű programokat.

A számítógépes rendszer szintjei - felhasználói programok

- felhasználói programok
  - pl. szövegszerkesztő (word processzor), adatbázis-kezelő, reptéri jegyfoglaló rendszer, egy játékprogram a mobiltelefonon
- rendszerprogramok
- hardver

1.2. Memória és az előtétszavak

A memória-hierarchia



A memóriát általános értelemben használják minden olyan eszközre, amely adatot képes tárolni. Gyakran viszont ezek közül csak azt értik csak alatta, amelybe a programok betöltődnek, ha futtatjuk azokat. (Ez utóbbit szokás fő memóriának vagy operatív tárnak is nevezni.) Ha azt modja valaki, hogy kevés a gépem memóriája, akkor ez utóbbit érti alatta.

A memória-hierarchiában lejjebb található szintek elérési ideje hosszabb, de az egységnyi adatmennyiségre (Bájtra) eső költsége kisebb.

A gépi kódú utasítások a regiszterekkel tudnak gyorsan számolni. Ha a memóriából kell behívni az adatokat, az jelentősen hosszabb időt vesz igénybe. Mivel gyakran a memóriából egymáshoz közeli adatos szükségesek, ezért a processzor egy köztes tárat, a cache-eket használja arra, hogy betöltse a várhatóan sorra kerülő adatokat.

Az előző bekezdésben említett memóriák addig tartják meg a tartalmukat, amíg a számítógép be van kapcsolva. Ha nincs áramellátás, a regiszterek, a cache és a memória tartalma elvész. A kiszámított értékeket, létrehozott táblázatokat, képeket ki kell írni egy biztonságosabb helyre. Ez tipikusan a merevlemez szokott lenni a számítógépek esetén.

Kutatások esetén rengeteg terabájt adat keletkezhet (pl. a CERN-ben) amelyeket nem szükséges mindig gyorsan elérnünk. Ezeknek a tárolását szalagon lehet legolcsóbban megoldani.

Mi a baj a jelenlegi előtétszavakkal?

- 1 Byte = 1 B = 8 bit (két tizenhatos számjeggyel leírható).

- Az adatátviteli sebességnél (1 kbit/s = 1000 bit/s) más a kilo jeletése, mint a memóriánál (1 kB = 1024 B)
- Más mérnöki területen más a kilo jelentése, mint a bájtok mellett.
- DVD-nél $4,7\ GB = 4,7 \cdot 10^9$ B.
- Memóriánál 1 GB $= 1024 \cdot 1024 \cdot 1024\ B = 1,074 \cdot 10^9$ B.

Az bináris előtétszavak

IEC - kb. Nemzetközi Villamosmérnök Bizottság

hagyományos	**IEC**	**értéke**	**elérés**
KB	KiB	$1024B = 2^{10}B = 1024B$	2%
MB	MiB	$1024^2B = 2^{20}B = 1048576B$	5%
GB	GiB	$1024^3B = 2^{30}B = 1,07 \cdot 10^9B$	7%
TB	TiB	$1024^4B = 2^{40}B = 1,10 \cdot 10^{12}B$	10%
PB	PiB	$1024^5B = 2^{50}B = 1,13 \cdot 10^{15}B$	13%
EB	EiB	$1024^6B = 2^{60}B = 1,15 \cdot 10^{18}B$	15%

- MiB szóban megabinari bájt vagy mibi bájt.
- Szemben az SI-vel a kilo jele is általában nagy K.

2. Az operációs rendszer

Forrás

- Tanenbaum–Woodhull: Operációs rendszerek, Panem, 1999
- Elearningen egy része szkennelve.

2.1. Az operációs rendszer fogalma

Az operációs rendszer

Erőforrásnak nevezzük a HW egyes részeit (nyomtató és más háttértárak, memória, CPU)
Az operációs rendszer

- megvéd a hardver bonyolultságától
- virtuális gépet alkot, melyet könnyebb megismerni/programozni
- erőforrásokat kezel
  - programok versenye a processzorért, memóriáért, I/O eszközökért
  - pl. több nyomtatás
  - pl. több felhasználó: HW és információigény (fájl írása olvasása)

2.2. Az operációs rendszer története

1. generációig (elektroncsövesig)

- nincs operációs rendszer
- 1. generációnál kapcsolótáblás majd lyukkártyás programozás

2. generáció (tranzisztoros)

- kötegelt rendszer lyukkártyán v. szalagon feladatok egymás után
- vezérlőkártyák: speciális kártyák, a parancsértelmezők előfutárai
- többnyire tudományos és műszaki feladatok (pl. időjárás-előrejelzés)
- többnyire FORTRAN és assembly programok

A többfeladatos rendszerek olyan rendszerek, amelyek esetén a számítógépen több program vár futásra. Ennek –többek között–az alábbi kettő változata van:

A multiprogramozás esetén csak akkor szakad meg egy program végrehajtása, és csak akkor indul el egy másik, amikor az első olyan műveletre (például nyomtatásra) vár, ami hosszabb ideig tarthat.

Az időosztásnál a felhasználó minden futó programja (folyamata) kap egy kis időszeletet, amíg a programja fut, majd másik folyamatra vált a rendszer, aztán harmadikra, ..., elsőre, másodikra, harmadikra, ... Ezek az időszeletek elég kicsik ahhoz, hogy a felhasználó úgy érezheti, hogy minden folyamata egyszerre fut.

A többfelhasználós rendszerek alapja is az időosztás. Ebben az esetben több felhasználó futtathat programokat ugyanazon a számítógépen.

4. generáció (LSI, μproc.)

- személyi számítógépek
- szoftvergyártás ipara
- MS-DOS, Windows 3.1 | Win95...
- Unix (AT&T, ingyen majd pénzért, jogvédve) MINIX (oktatásra, Tanenbaum) $\Rightarrow$ Linux (Linus Torvalds)

A Windows 3.1 még nem tekinthető egy önálló operációs rendszernek, hanem egy MS-DOS-on futó grafikus felületű programnak. A Windows 95 már igazi időosztásos működésű grafikus felületű operációs rendszer.

Az 1969-ben kifejlesztett Unix-okat kezdetben ingyenesen használhatták az egyetemek, megismerhették a forráskódját.

A Linuxot az oktatási célra készült MINIX (mini Unix) kódjából indította útjának Linus Torvalds svéd anyanyelvű finn (akkor még) egyetemi hallgató 1991-ben. Erre épülnek különböző Linux terjesztések, mint a főiskolánk szerverein futó Debian-ok, vagy az asztali gépeken könnyű használhatóságot kitűző Ubuntu. `http://ubuntu.hu`

4. generáció – hálózat

Számítógéphálózatok növekedése (1980–)
Hálózati operációs rendszer

- egyszerű bővítések: hálózati csatoló, bejelentkezés távoli gépre, fájlátvitel kezelése

Osztott operációs rendszer

- több processzor vagy számítógép
- kezeli hogy mi melyiken fut/tárolódik
- bonyolultabb folyamatütemező algoritmus
- GRID: a másik processzor másik földrészen is lehet

A CERN adattermelése

Az európai részecskefizikai kutatóközpont, a CERN, adatainak feldolgozásához nem elegendő a CERN számítógépközpontjában található rengeteg gép. A számításokat egy olyan GRID végzi, amelyhez máshonnan is kapcsolódnak számítógépek, mint például más európai, amerikai és japán felsőoktatási intézmények és kutatóintézetek számítógépei.

A GRID felépítése

A GRID-et egy köztes rétegnek (Middleware-nek) nevezett program működteti. Feladata a futásra váró alkalmazásoknak megkeresni a futásra leginkább megfelelő helyet, ehhez tudnia kell, hol találhatóak a futáshoz szükséges adatok, melyik gép mekkora szabad számítási kapacitással rendelkezik (nem túl terhelt CPU-t kell keresni)... Emellett ellenőrizni kell a felhasználók jogosultságait, mérni kell, ki mennyit használ illetve ajánl be.

A köztes réteg alatt helyezkedik el az erőforrás réteg, amelyek részei:

háttértárak (merevlemez, szalag (CERN-ben) magnókazetta méretű 500GB-os szalagok vannak, amiket egy robotkar emel le és olvas merevlemezre, ha szükség van rá)

számítógépek, amelyek a számítást végzik

különböző érzékelők és detektorok (pl. részecskefizikai detektorok, távcsövek, időjárás érzékelők)

A köztes réteg felett található az alkalmazási réteg, amelyből a feladatot a GRID-nek továbbítják.

A számítógépeket bármikor ki lehet venni a rendszerből és hozzá lehet adni anélkül, hogy a rendszert újra kellene indítani.

A kutatási intézetek és felsőoktatási intézmények erőforrásait ki lehet használni akkor is, amikor ott helyben nem lenne dolog.

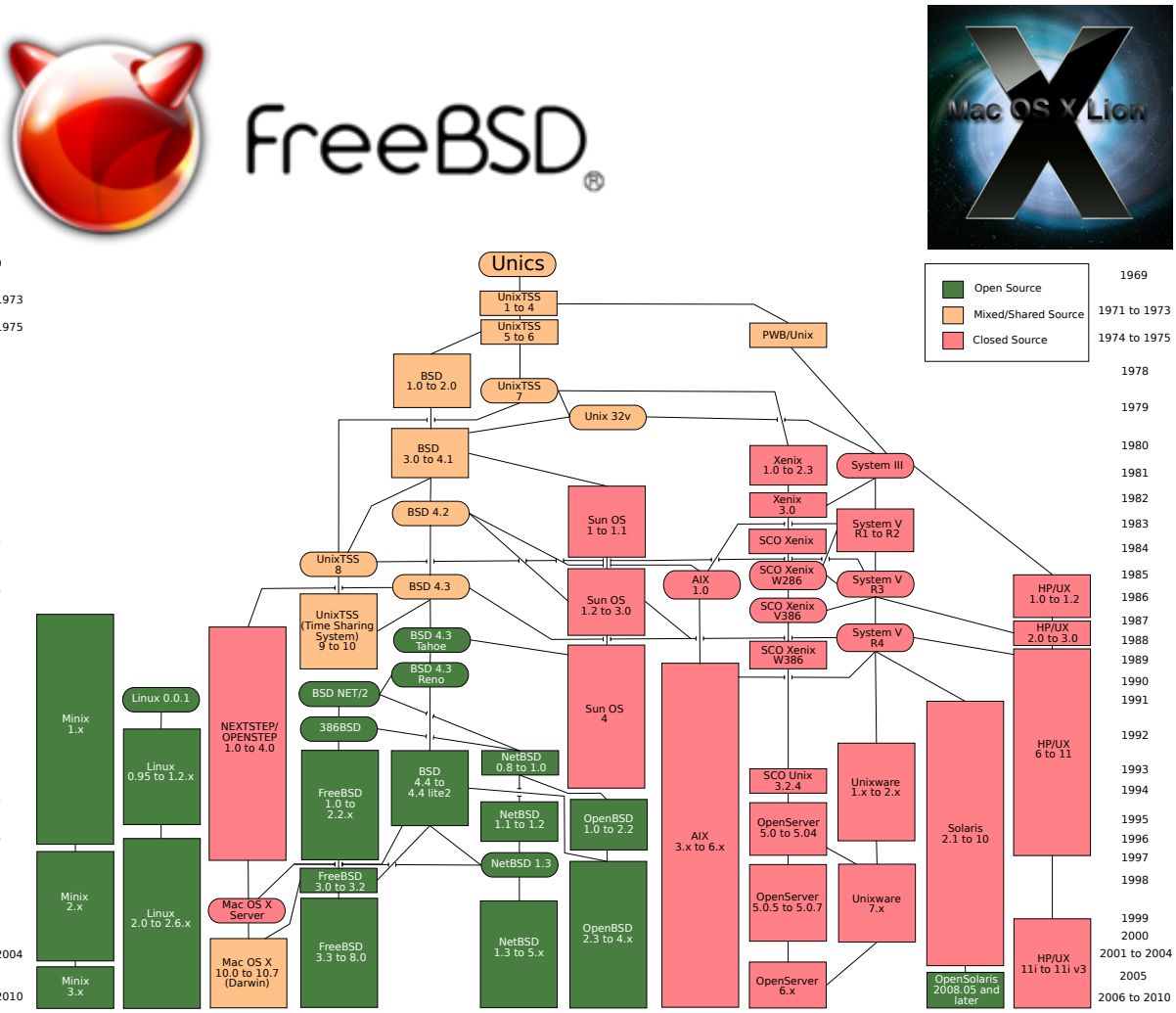
A számítógépeket bármikor ki lehet venni a rendszerből és hozzá lehet adni anélkül, hogy a rendszert újra kellene indítani.

A kutatási intézetek és felsőoktatási intézmények erőforrásait ki lehet használni akkor is, amikor ott helyben nem lenne dolog.

A Unix és a Unix-szerű operációs rendszerek

A Unix kialakulása

- 1969 Unix AT&T
- 1972 C programozási nyelv, Unix-ot újraírták C-ben.
- Utódai: (Open)Solaris (SUN), AIX (IBM), HP-UX (HP), BSD, Mac OS X
- 1984 GNU projekt egy teljes Unix-kompatibilis szoftverrendszer, Szabad Szoftver Alapítvány (FSF)
- 1991 Linux kernel (⇒ Android)



A Unix-ot 1969-ban fejlesztették ki AT&T vállalatnál. 1972-ben ugyanitt fejlesztették ki a C programozási nyelvet, és a korábban assemblerben írt Unix-ot újraírták C-ben. Kezdetben szabadon terjeszthető volt forráskóddal együtt, majd üzletet látva benne zárt kóddal pénzért árulták. Több változata lett System V (V=öt), BSD (Berkley Software Distribution), HP-UX, SUN Solaris, Mac OS X. Ezek egy része utóbbi később szabad szoftverré vált: *BSD, OpenSolaris.

A Szabad Szoftver Alapítvány (FSF) által 1984-ben indított GNU projekt célja egy teljes Unix-kompatibilis szoftverrendszer kifejlesztése volt. Ebből származik a Linuxra épülő terjesztések fordítóinak, szövegszerkesztőinek, parancsértelmezőinek (shell-jeinek) jelentős része. És az olyan Unix-os parancsok nyílt forráskódú változata, mint például az ls, cp parancsok (listázás, másolás).

1987-ben egy egyetemi tanár, Andy Tanenbaum, kifejlesztett oktatási célokra egy minimális operációs rendszert, a MINIX-et. Ennek teljes forráskódja megtalálható Tanenbaum: *Operációs rendszerek* című könyvében.

Ezt használta fel Linus Torvalds, svéd anyanyelvű finn egyetemi hallgató egy operációs rendszer kernelének a kifejlesztéséhez, melyet később Linuxnak neveztek el és azóta is Linus fejleszti. A munkát 1991-ben kezdte.

Precíz szóhasználatban a Linux a kernelt jelenti. Teljes operációs rendszerré a Linuxot a GNU-s programok teszik, ezt az operációs rendszert GNU/Linuxnak szokták hívni. A különböző Linux terjesztések ezeket a programokat szoftvercsomagokba szervezik és csomagkezelő programokat használnak a szoftvercsomagok telepítésének egyszerűbbé tételéhez.

A GNU/Linux operációs rendszereknek nem részei az ablakozó rendszerek (pl. IceWM, FluxBox) és a összetettebb grafikus felhasználói felületek (GNOME, KDE). A telepítés lehetséges csupán karakteres felülettel, de beállítható a rendszer úgy is, hogy a rendszer grafikus beléptetőprogrammal indul, a Windowshoz hasonlóan.

A GNU/Linux rendszerek és a Unix-ok között jelentős hasonlóság van, amit több szabvány, például a POSIX nagymértékű betartása tesz lehetővé. Aki egy Unix-rendszert ismer, az nem fog elveszni a Linuxban és fordítva.

A BSD-változatok (FreeBSD, NetBSD, OpenBSD, PC-BSD,...), melyek szintén szabad szoftverek. Amíg a Linuxot több összetevőből rakják össze (GNU projekt, Linux-kernel), addig a BSD-k teljes alaprendszere (kernel+operációs rendszer) együtt kerül fejlesztésre így jobban összecsiszolt. Végül két érdekesség: a Mac OS X operációs rendszer, az Apple Macintoshának 2002-ben bevezetett operációs rendszere szintén BSD alapú, az Android pedig a Linux leszármazottja.

Linux, Ubuntu



- *Az Ubuntu* a *GNU/Linux* operációs rendszer egyik disztribúciója
- `ubuntu.hu`
- A Debianból származó *deb* szoftvercsomagokat használ
- A deb fájlok *optikai diszkről* vagy *Internetes tárolókból* érhetőek el.
- apt csomagkezelő rendszer: telepítés függőségekkel együtt, eltávolítás, frissítés, keresés

A telepített Ubuntu később hasznos lehet az „Operációs rendszerek” laborgyakorlatához, és a szabadon választható „Linux alkalmazása a gyakorlatban” illetve „Összetett hálózatok vizsgálata” tárgyakhoz.

Régebbi, kisebb memóriával rendelkező gépekre a Lubuntu változatot ajánlom, az újabbakra az Ubuntu vagy Kubuntu változatot. Bármelyik telepítés mellé fel lehet rakni a másik kettő változatot is utólag.

2.3. Szoftverekkel kapcsolatos fogalmak

- Szabad szoftver (Free Software) olyan szoftver, amely szabadon használható, tanulmányozható és módosítható, és a módosított változat is szabadon továbbadható. Ezekhez szükséges a forráskód ismerete. Nem feltétlenül jelent ingyenességet. Támogatást lehet hozzá pénzért adni, vagy lehet pénzt kérni azért, hogy valami kiegészítést ír valaki a kódhoz, de megszünteti a készítőtől való függőséget, mert ha nem megfelelő nekem amit és amilyen áron csinál nekem a készítő, átadhatom a munkát másnak.

- Public Domain: olyan programok, amelyeket szabadon lehet használni, és nem csak próbaidőre. Ezek nem feltétlenül szabad szoftverek a fenti értelemben.

- Nyílt forrású (open source) program: olyan szoftver, amelynek a forrását szabadon elérhetővé teszi a gyártó. Ez sem feltétlenül szabad szoftverek, ha a fenti feltételek nem teljesülnek. Egy cég például szabadon elérhetővé teheti a program forrását úgy is, hogy nem enged a kódban változtatást, és pláne a változtatott kód továbbadását. Lehet erre az az oka, hogy a vásárló megbizonyosodhasson arról, hogy a program nem tartalmaz hátsó kapukat, amelyen információk áramlanak ki.

- Dual Boot: A számítógép olyan állapota, amikor kétféle rendszer (pl. Windows és Linux) indítása is lehetséges.

- Live CD: olyan CD, amelyről telepítés előtt kipróbálható a telepítendő operációs rendszer. Ilyenek például az Ubuntu telepítő lemezei. Természetesen ilyenkor nem olyan gyors a rendszer futása, mintha egy külön partícióra telepítenénk, és az adataink is elvesznek, hiszen azok nem kerülnek a Winchesterre, csak a memóriába. Viszont kipróbálható, hogy az adott hardver milyen képességét képes kihasználni a rendszer.

- Virtuális gép (VMware, Virtualbox): egy lehetőség, hogy egy operációs rendszer alá másikat telepítsünk. A fenti kettő program Windows és Linux alatt is lehetőséget teremt, hogy egy másik operációs rendszert futtassunk az eredeti rendszer egyik ablakjában.

2.4. Az operációs rendszer alapfogalmai

Alapfogalmak

- Folyamat (processzus, process)

- Fájlrendszer

- Parancsértelmező (héj, shell)

Folyamatok

Folyamat: végrehajtás alatt álló program, amely rendelkezik

1. címtartománnyal (memóriaszelet) program, adat, verem
2. regiszterkészlettel (CPU-ban) ezek tárolják a számolás során az adatokat. Két speciális regiszter: utasításszámláló, veremmutató

Időosztásos rendszerben időnként megszakítás van, ekkor

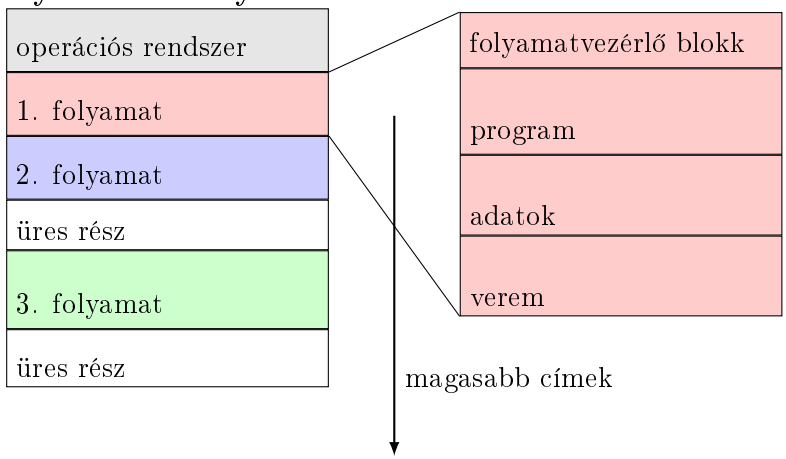
- menteni kell a regiszterek értékeit
- amikor újra erre a folyamatra kerül a sor, akkor vissza kell ezeket tölteni

Az utasításszámláló arra a címre mutat, ahol a program végrehajtása éppen tart. Legtöbb utasítás után a következő utasításra fog mutatni az utasításszámláló. Kivételt jelentenek az ugró utasítások.

Kiegészítő anyag. A gépi kódban van például olyan utasítás, amely hogyha egy adott regiszter nem nulla, akkor ugrik egy adott helyre (máshol folytatja a végrehajtást), különben a következő utasításon folytatja a végrehajtást.

Ilyen módon lehet ciklust szervezni: az adott regiszterbe belerakom, hányszor kell végrehajtani a ciklust. Minden egyes ciklusban csökkentem a regiszter értékét. Ha nem nulla a regiszter értéke, akkor vissza kell ugrani a ciklus elejére.

A folyamatok elhelyezkedése a memóriában



A központi feldolgozó egységben (CPU) több néhány bájtos kis tárolóegység található, az úgynevezett regiszterek. Ezekben találhatóak többek között a számolások eredményei. A regiszterek értékeit sokkal gyorsabban eléri a processzor, mint a memóriában található értékeket.

A regiszterek egyike, az utasításszámláló mutat arra a helyre, ahol a következő végrehajtandó utasítás található a memóriában. Általában egy utasítás végrehajtásakor ez eggyel növekszik. Kivételt jelentenek az ugró utasítások. Egy processzor rendelkezhet például olyan ugró utasítással, amely egy regiszter nullától különböző értéke esetén ugrik egy adott memóriacímre, és onnan folytatja a program futtatását. (Ezzel megvalósítható egy adott számszor lefutó ciklus. A számot beírjuk a regiszterbe, minden ciklusban csökkentjük, és ha nem nulla visszaugrunk. Ha nulla, akkor végeztünk a ciklussal, folytatjuk tovább a következő utasítással.)

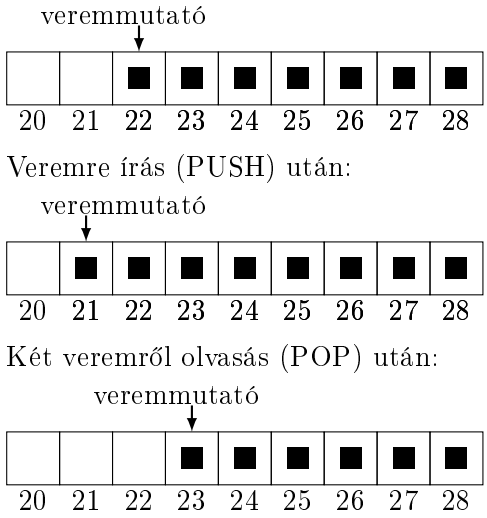
Végrehajtáskor a háttértárolóról (általában merevlemezről) a memóriába kerül a program, és a hozzá tartozó adatok. A program a futása során néha veremre ír ki adatokat. A verem egy olyan tárrész a memóriában, amelyből a legutoljára beírt adatokat tudjuk legelőször kivenni, ezért nevezzük LIFO-elvű tárolónak (Last In First Out = először be utoljára ki). A végrehajtás alatt álló programnak (folyamatnak) egy memóriarészt ad az operációs rendszer. Tipikusan a memóriaszelet kisebb címeinél található a program és a hozzá tartozó adatok, és a legnagyobb című helytől a kisebbek felé tárolja az adatokat a verem. Azt, hogy hová kell a következő értéket írni a verembe, a veremmutató nevű regiszter tárolja.

Kiegészítő anyag. A verem célszerű megoldás úgynevezett alprogramok meghívásakor, amikor a korábban végrehajtott programrész regisztereinek értékeit el kell menteni, és azt is, hogy az alprogram végrehajtása után melyik memóriacímre kell visszatérni a program futásának.

Ha az alprogramok hívásakor veremre rakom (a többi regiszterrel együtt) azt, hogy hol tartott a program (azaz az utasításszámláló értékét) az alprogram hívása előtt, akkor könnyen tudom kezelni azt, hogy honnan kell folytatni az eredeti programot, amikor az alprogram végére érek. Az sem okoz gondot, ha többször meghívom az alprogramból saját magát, mert a veremből mindig a legutolsó alprogram-hívás regiszterértékeit fogjuk visszakapni.

A verem elnevezése abból ered, hogyha verembe pakolok, akkor csak azt tudom kivenni, amit legutoljára raktam be, ami a verem tetején van. (Ellentétben például a szekrény fiókjaival, ahol például akármelyikből kiszedhetem bármikor tetszőleges sorrendben, ami benne van.) Azzal, hogy a memóriában a kisebb címek felé halad a verem, ha beleírok, egy kissé fejreáll a helyzet: a verem „tetején" lévő szám van a legkisebb című helyen. Ennek ellenére szokás az utolsónak beírt számot a verem tetején lévő számnak nevezni.

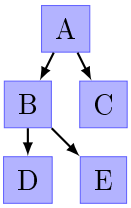
Veremre írás-olvasás



A veremmutató arra a címre mutat, ahol a verem legutolsónak berakott eleme található. Kétféle veremművelet létezik. Az egyik (PUSH) a veremre ír egy számot. Ilyenkor a veremmutató helyére kerül a szám, és a veremmutató értéke eggyel csökken. A másik utasítás (POP) egy számot tölt be a veremről az egyik regiszterbe, és a veremmutató értéke eggyel növekszik.

A fenti ábrán a fekete kis négyszögek jelölik azokat a helyeket, ahol már adat van, az egyes lépések után. (A valóságban a 28-nál sokkal nagyobb memóriacímeken található a verem „alja".)

Szülő- és gyermekfolyamatok



- B és C az A gyermekfolyamatai
- A a B és C szülőfolyamata
- egy egyedi szám, a folyamat-azonosító (PID, process identifier) tartozik minden folyamathoz
- felhasználó-azonosító (UID, user identifier) tartozik minden folyamathoz (ki indította), a gyermekfolyamatok öröklik

Szülő- és gyermekfolyamatok

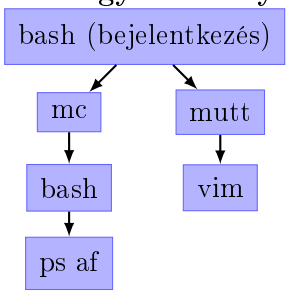
```
  PID TTY      STAT   TIME COMMAND
  14691 pts/0    Ss     0:00 -bash
  14758 pts/0    T      0:00  \_ mc
  14760 pts/2    Ss+    0:00  |   \_ bash (...)
  14920 pts/0    T      0:00  \_ mutt
  14927 pts/0    T      0:00  |   \_ vim (...)
  14937 pts/0    R+     0:00  \_ ps af
(...)
```

bash: egy parancsértelmező, a Linuxok alatt elterjedt
mc (Midnight Commander): fájlkezelő, a Norton Commanderhez hasonlít
vim: egy szövegszerkesztő (text editor), amely itt a levelet szerkeszti
ps: a folyamatok listázására alkalmas program Unix/Linux alatt a `f` (forest) kapcsolója alkalmas a folyamatok hierarchikus (alá-fölé rendelt) megjelenítésére.

Bejelentkezéskor rögtön a bash-be kerültem, ahonnan megnyitottam a mutt parancsot, és onnan egy levelet kezdtem írni a vim szövegszerkesztővel. Ezt háttérbe küldtem, majd megnyitottam az mc fájlkezelőt, ami rögtön megnyit magán belül is egy bash-t, ebben írtam be a `ps af` parancsot. A `ps af` parancs adta ki a fenti kimenetet.

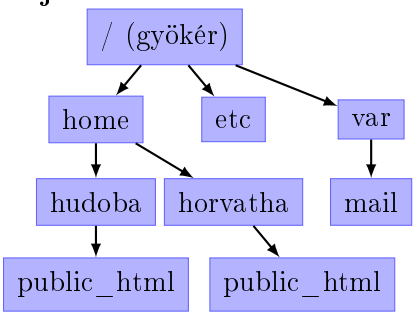
Mindegyik folyamatnak egyedi azonosítója (process id, PID) van.

Szülő- és gyermekfolyamatok



A mappa (folder) – első közelítésben – a könyvtárak (directory) grafikus megjelenítése. Akit precízebben érdekel: `http://superuser.com/questions/169457/directory-vs-folder`

Fájlrendszer Unix és Linux rendszerekben



útvonalnév lehet

- abszolút: gyökérkönyvtárból pl. `/home/horvatha`
- relatív: pl. `../hudoba/public_html`

A felhasználók saját könyvtárai tipikusan a /home könyvtárban találhatóak. Szülőkönyvtár, alkönyvtár fogalma.

Szóhasználat: `home` az `horvatha` könyvtár *szülőkönyvtára*, az `horvatha` a `home` *alkönyvtára*.

A Unix és Linux rendszerek egyetlen hierarchikus könyvtárrendszerrel rendelkeznek, nincsenek külön meghajtók (C:) mint a Windows esetén. A csatolt fájlrendszerek például a /media könyvtárban helyezkednek el, (pl. /media/cdrom).

A Unix rendszerekben – a webcímekhez hasonlóan – az útvonal tagjait `/` választja el a Windowsos `\` helyett.

Tegyük fel, hogy kezdetben a `/home/hudoba` könyvtárban vagyok, amit az ábrán is láthatok. A `..` (Windows alatt is) mindig az aktuális könyvtár szülőkönyvtárát jelenti, tehát a `../horvatha/public_html` azt jelenti, hogy az aktuális könyvtárból kilépek a szülőkönyvtárba, a `/home` könyvtárba, majd belépek a horvatha könyvtár `public_html` alkönyvtárába, tehát a `/home/horvatha/public_html` könyvtárba jutok (ahol egyébként az horvatha felhasználó weboldalai helyezkednek el, amit a `www.arek.uni-obuda.hu/~horvatha` oldalról érek el).

Parancsértelmező (shell)

- a billentyűzetről beírt parancsokat hajtja végre
- saját vezérlési szerkezetekkel rendelkezik (ciklus, feltételes elágazás)
- fájlban összegyűjtött parancsok (shell script) végrehajtására is képes
- fejlettebbeknél automatikus kiegészítés, korábbi parancs visszahívása

2.5. Összefoglalás

Összefoglalás

Fontosabb fogalmak

- operációs rendszer

- rendszerprogram
- felhasználói program
- erőforrások
- hardver
- időosztás
- többfeladatos és többfelhasználós rendszer
- folyamat
- parancsértelmező = shell

3. Linux a gyakorlatban

Manager Informatika laborhoz és az informatikus szabadon választható Linux alkalmazása gyakorlathoz. Az Informatikai rendszerek alapjaihoz nem szükséges.

rwx bitek
r - read, w - write, x - eXecute

```
$ ls -l /etc
összesen 1600
drwxr-xr-x  8 root  root   4096 2007-04-18 05:45 acpi
-rw-r--r--  1 root  root   2657 2007-04-18 05:40 adduser.conf
-rw-r--r--  1 root  root     47 2007-09-30 16:49 adjtime
-rw-r--r--  1 root  root     47 2007-05-11 17:31 aliases
drwxr-xr-x  2 root  root  20480 2007-06-25 21:50 alternatives
-rw-r--r--  1 root  root    395 2007-03-05 07:38 anacrontab
drwxr-xr-x  7 root  root   4096 2007-04-18 05:45 apm
drwxr-xr-x  4 root  root   4096 2007-06-05 22:49 apt
(...)
```

3. és 4. oszlop tulajdonos és csoport, itt mindkettő neve root.

A Unix és Linux rendszereken a root felhasználó a rendszer legfőbb adminisztrátora. Korábban az ő saját könyvtára a gyökérkönyvtár (root) volt, innen kapta a nevét.

rwx bitek

jog	*fájlra*	*könyvtárra*
r - read	olvasható	listázható
w - write	írható	létrehozható/törölhető fájl benne
x - eXecute	végrehajtható	be lehet lépni (`cd` paranccsal)

```
-rw-r--r--  1 root   root       969 2007-04-18 05:40 group
drwxr-xr-x  4 root   root      4096 2007-06-05 22:49 apt
```

nem könyvtár	*felhasználó*	*csoport*	*mások jogai*	*(...) fájlnév*
-	rw-	r--	r--	(...) group

könyvtár	*felhasználó*	*csoport*	*mások jogai*	*(...) fájlnév*
d	rwx	r-x	r-x	(...) apt

A /etc/group fájl tartalmazza, hogy milyen csoportok találhatóak az operációs rendszerben, és azokban kik vannak benne. Ez a root tulajdonában van, aki olvashatja és írhatja is (rw). Emellett a root csoportnak és minden egyéb felhasználónak is olvasási joga van rá.

Végrehajtási joga senkinek nincs, hiszen ez nem egy végrehajtható fájl, hanem egy információkat tartalamzó szövegfájl.

A /etc/apt egy könyvtár, amelybe mindenkinek joga van belépni (x) és listázhatja mindenki (r), de fájlokat törölni és írni a könyvtárban csak a tulajdonosnak (itt a root) van joga.

Határozzuk meg a következő sorból, mi a fájlok/könyvtárak neve, fájl-e vagy könyvtár, ki a tulajdonosa, csoportja, és kinek milyen joga van!

```
-rw-rw-r-- 1 hudoba   fizika 1481 2006 nov 10  CERNLogo.png
drwxrwxr-x 9 horvatha fizika 4096 okt 30 16.01 diakmuhely
-rwxr-xr-x 1 root     root  92312 2008 ápr  4  /bin/ls
drwxrwxr-x 2 root     mail   4096 nov 23 19.25 /var/mail/
-rw------- 1 horvatha mail 47099518 nov 23     /var/mail/horvatha
```

Pár alapparancs

- `ls` (list) listázza a pillanatnyi könyvtár tartalmát
- `ls -l fájlnév` listázza a fájl jogosultságait és más tulajdonságait a korábbi fólia szerint
- `ls -l [könyvtár]` listázza a könyvtár összes fájlját az előzőhöz hasonlóan; ha nincs könyvtár megadva, az aktuális könyvtárét
- `pwd` (print working directory), kiírja, melyik könyvtárban vagyunk
- Linux parancssorban mindig elérhető a nano szövegszerkesztő. `nano [fájlnév]` (ha nincs fájlnév, akkor mentéskor kell megadni az új fájl nevét)

(A nanoban alul megtalálhatóak a legfontosabb billentyű-kombinációk jelentései; felül látható a fájlnév és hogy módosításra került-e a fájl az utolsó mentés óta.

Egy nagyobb tudású szerkesztő a Vim, a használatához viszont többet kell tanulni. Ha grafikus felület is rendelkezésünkre áll, akkor akár a Python grafikus felületű szerkesztőjével, az IDLE-vel is szerkeszthetünk szövegfájlokat.)

Könyvtárak kezelése

- `cd [útvonal]` (call directory), könyvtárba belépés, ha nincs útvonal, a saját-könyvtárba
- `cd ~/linux/segedlet`
- `mkdir útvonal` (make directory), könyvtár létrehozása
- `mkdir pistike`
- `rm -r könyvtár` (remove) törli a könyvtárakat a benne levő könyvtárakkal és fájlokkal együtt (-r = rekurzívan)
- `rm -r pistike`

Fájlok jogosultságainak megváltoztatása

- `chmod a+r fájl` (change mode all + read) Megadja a jogosultságot, hogy bárki olvashassa a fájlt. (Weboldalaknál, majd kell ilyen jogosultság.)
- `chmod a+x fájl` (change mode all + eXecute) Megadja a jogosultságot, hogy bárki végrehajthassa a fájlt. (Programok végrehajtásához = futtatásához, szükséges.)
- `chmod a-x fájl` (change mode all - eXecute) Megvonja a jogosultságot mindenkitől, hogy végrehajthassa a fájlt.

Fájlok/könyvtárak másolása és mozgatása/átnevezése

- `cp fájl cél` (copy) a fájlt átmásolja a célba. Ha az könyvtár, akkor bele, ha fájlnév, akkor arra a névre.
- `mv fájl cél` (move) a fájlt átmozgatja a célba (azaz az eredeti megszűnik). Ha az könyvtár, akkor bele, ha fájlnév, akkor arra a névre. Fájl helyett könyvtárral is működik. Átnevezhetek ezzel egy fájlt/könyvtárat (`mv réginév újnév`).
- Az fájl (könyvtár) nevénél használhatjuk a *-ot tetszőleges fájlrész helyett.
- `cp *.jpg images` Ha images egy könyvtár, az összes .jpg végű fájlt átmásolja bele.
- `cp -r könyvtár cél` (rekurzívan) A könyvtárat átmásolja. Ha a cél létező könyvtár, akkor bele, ha még nem létező név, akkor arra a névre.
- `rm fájl` fájl törlése.

Feladatok

- Listázzuk a public_html illetve /tmp könyvtárak összes fájlját és alkönyvtárát. Ki a tulajdonosuk? Milyen jogokkal?
- Másoljuk a szamtech/info/web könyvtár összes fájlját és alkönyvtárát a saját `public_html` könyvtárbeli alkönyvtárunkba.
- Lépjünk be a public_html könyvtárba!
- Hozzunk létre egy könyvtárat, amelynek a neve a saját NEPTUN-kódunk!
- Másoljuk át a public_html könyvtár valamelyik fájlát a most létrehozott könyvtárunkba!

Belépés a django szerverre Putty-val

Böngészőbe a következő címet kell megadni: `http://django.arek.uni-obuda.hu/~infor`
Putty-ba ezek kellenek:

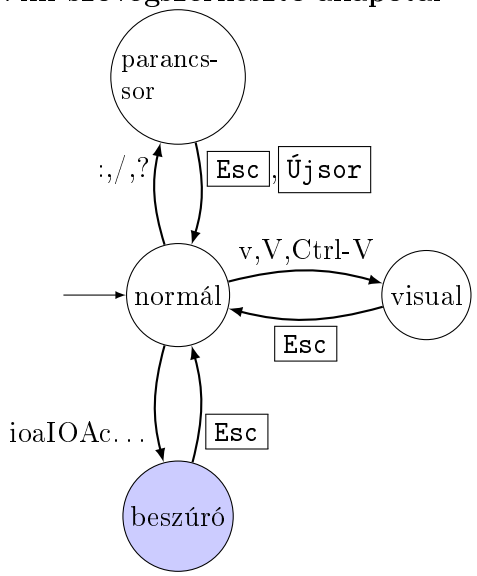
- Server name: django.arek.uni-obuda.hu
- Port: 122
- Translation UTF-8 kódolás

A jelszót ne tároljuk!
Open után első belépéskor megjelenik egy ablak (Security alert), ott Igent kell nyomni.
Utána a felhasználónevünket majd a jelszavunkat kell megadni. A jelszót vakon kell begépelni, semmit nem látunk a képernyőn írás közben.

Vim szövegszerkesztő

- A nano helyett érdemes a nagyobb tudású, de több tanulást igénylő vim szövegszerkesztőt használni.
- Több szöveget egyszerre tarthatunk benne nyitva, mindegyiket egy-egy pufferben. A pufferek között viszonylag könnyű sorokat másolni és mozgatni.
- Indításkor *normál módban* van. Ebben a módban a betűknek speciális jelentésük van.
- A *beszúró módba* általában az i, vagy o betűvel jutunk. Az utóbbi új sort nyit. Escape gombbal térhetünk vissza.
- A *parancs módban* kereshetünk (/ vagy ?) vagy valamilyen parancsot adhatunk (:), kiírhatjuk a puffer tartalmát egy fájlba, megnyithatunk új fájlt, segítséget kérhetünk, átállíthatunk valamilyen beállítást... Az `Újsor` után keres vagy végrehajtja a parancsot és visszatér normál módba.

A Vim szövegszerkesztő állapotai



Normál mód, mozgások

`^` vagy `0`	sor eleje
`$`	sor vége
w vagy W	következő szó
b vagy B	előző szó
`{ }`	előző és következő üres sorig
`G`	fájl végéig
`22G`	22. sorra
`gg`	fájl elejéig

Nyilak, `End`, `Home` általában működnek.
Némelyiknél ismétlés is van. `5}`, `2w`

Normál mód, cselekvések

d	töröl
y	másolásra kijelöl
c	megváltoztat (töröl+beszúró mód)
p	beilleszt (aktuális betű/sor után)

Általános forma ismétlés+cselekvés+mozgás. (Ismétlés elhagyható. Beillesztésnél nincs mozgás.)

Pl. `5dw` öt szót töröl, `3p` háromszor beilleszt, `c$` sor végéig töröl és beilleszt.

Speciális eset: sort töröl dd öt sort töröl 5dd, hasonlóan yy és cc.

Parancssor

`:n név`	új puffert nyit (new/next)
`:sp név`	új puffert nyit ablakot felosztva (split)
`:w`	ment (kiírja a puffer tartalmát, write)
`:w név`	ment adott néven
`:q`	kilép a pufferből (csak ha mentve volt)
`:wq`	kiírja és kilép (write+quit)
`:q!`	kilép mentés nélkül
`:ls`	listázza a megnyitott puffereket
`:b sorszám`	adott sorszámú pufferre ugrik (egybe írható)
`:b név`	adott nevű pufferre ugrik
`:!parancs`	shell parancs végrehajtása
`/minta`	adott mintát keres lefelé
`?minta`	adott mintát keres felfelé

A keresésnél a következőt az adott irányba az n-nel, a másik irányba N-nel kereshetjük meg.

Példák parancsorra

`:!ls -l` – listáz, mintha Linux-parancssorba írnánk.

`:n .bashrc` – megnyitja a .bashrc fájlt.

`:sp .bash_history` – megnyitja a .bash_history fájlt úgy, hogy felosztja az ablakot alsó és felső részre.

`:q` – kilép az aktuális ablakrészből

`:ls` – kilistázza a puffereket.

`:b2` – a kettes pufferre (angolul buffer) lép.

`:q` – kilép (csak, ha nincs mentetlen puffer)

Minden visszavonható – u

Könnyű nagy változást csinálni a vimmel, de normál módban u-val minden visszavonható.

A vimtutor és egyebek

A vim elsajátításához érdemes a vimtutor parancsot beírni. Ez végigvezet a Vim megismerésén.