

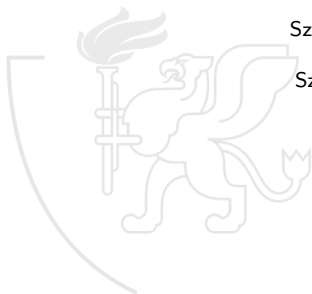
Programozás Alapjai

Dr. Gergely Tamás
Dr. Jász Judit

Szegedi Tudományegyetem
Informatikai Intézet
Szoftverfejlesztés Tanszék

2023

(v0911)



- 1 **Bemutakozás**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2 **Linux**
 - Alapfogalmak
 - Linux parancsok
 - Linux shell
 - Felhasználók
 - Hálózat
- 3 **Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszoátlag adott elemszámra
 - Csúszoátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4 **Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai
- 5 **Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.
- 6 **Folyamatábra és struktúradiagram**
- 7 **Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- Sztringek
 - Pointerek és tömbök C-ben
 - Rekord adattípus
 - Függvény pointer
 - Halmaz adattípus
 - Flexibilis tömbök
 - Láncolt listák
 - Típusokról C-ben
- 8 **IO**
 - Alapok
 - Adatállományok
 - 9 **C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok
 - 10 **Gyakorlati kérdések**
 - Memóriahasználat
 - Gyakori C hibák
 - where.c felboncolva

- 1 **Bemutakozás**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2 **Linux**
 - **Alapfogalmak**
 - Linux parancsok
 - Linux shell
 - Felhasználók
 - Hálózat
- 3 **Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszoátlag adott elemszámra
 - Csúszoátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4 **Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai
- 5 **Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.
- 6 **Folyamatábra és struktúradiagram**
- 7 **Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- Sztringek
 - Pointerek és tömbök C-ben
 - Rekord adattípus
 - Függvény pointer
 - Halmaz adattípus
 - Flexibilis tömbök
 - Láncolt listák
 - Típusokról C-ben
- 8 **IO**
 - Alapok
 - Adatállományok
 - 9 **C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok
 - 10 **Gyakorlati kérdések**
 - Memóriahasználat
 - Gyakori C hibák
 - where.c felboncolva

Linux alapismeretek

Miért?

- Egy informatikus számítógépekkel dolgozik, így nem engedheti meg magának, hogy ne tudja kezelni azt.
- Konkrétabban: a gyakorlaton programokat kell majd írni gépek segítségével, és ezeket tudni kell kezelni.
- Minél több rendszert ismer valaki, annál átfogóbb képet kap, és könnyebben ismerkedik újakkal. (Van Windows is másik kurzuson.)



Miért a Linux operációs rendszer?

- Sok helyen nagyon elterjedt:
 - 2017. november 1. óta a Top 500 mainframe 100%-án linux alapú rendszer fut.*
 - Beágyazott rendszerekben (SetTopBox-ok, TV-k, routerek, Raspberry, Arduino) elterjedt.
 - Az Android is Linux (és Java) alapokra épül.
- A Unix alapok miatt szorosan összefonódik a C nyelvvel.
- A parancssor-orientáltság miatt egyszerűbben lehet programokat írni (kisebb a belépési küszöb).
- A root nem „csak” egy adminisztrátor.

*<https://www.top500.org/statistics/details/osfam/1/>

„A Linux csak egy kernel, az operációs rendszer a GNU.”

– Richard M. Stallman

- A GNU/Linux egy POSIX szabványokat követő Unix-szerű operációs rendszer
 - GNU = „**G**NU's **N**ot **U**nix”
 - Többtaszkú, többfelhasználós, virtuális memóriával, védett üzemmóddal, korszerű memóriakezeléssel, megosztott programkönyvtárakkal, demand paging mechanizmussal, széles körű TCP/IP hálózati támogatással, stb.
- És ez mit jelent?
 - Most annyit, hogy egy normálisan használható operációs rendszer . . .
 - . . . de különböző kurzusokon (pl. operációs rendszerek) mindegyik kifejezés el lesz magyarázva.

- Olyan programrendszer, amely közvetítő szerepet tölt be a számítógép hardver erőforrásai és a felhasználó között.
- Főbb funkciói:
 - Programok betöltése és végrehajtása
 - Erőforrások elosztása
 - Input/output műveletek végzése
 - Háttértárakon tárolt adatrendszerek kezelése
 - A felhasználó által kiadott parancsok értelmezése és végrehajtása
 - A működés közben fellépett hibák lekezelése



- Az adatok (szövegek, képek, hangok, programok, stb.) hosszú távú tárolására a háttértárak szolgálnak.
- Tárolási egységek
 - bit** A legkisebb tárolási egység, 0 vagy 1 értékkel.
 - bájt** 8 bitnyi információ $2^8 = 256$ értékkel.
 - fájl** A szorosan egymáshoz tartozó, egy összetett adatot alkotó bájtokat a háttértárakon egy fájlban tároljuk.
 - könyvtár** A fájlokat a háttértárakon könyvtárakban tároljuk.



- A háttértárolón tárolt adatok logikailag egy fa szerkezetben vannak tárolva.
 - Gyökér (root) könyvtár
 - Alkönyvtárak
 - Fájlok
 - Aktuális könyvtár



fájlrendszer	útvonal
/	/
+ bin	/bin
+- bash	/bin/bash
+- cp	/bin/cp
:	
+ boot	/boot
+ dev	/dev
+ etc	/etc
+ home	/home
+- gertom	/home/gertom
+- .bashrc	/home/gertom/.bashrc
+- gyakorlat	/home/gertom/gyakorlat
:	
+- h531674	/home/h531674
:	
+ lib	/lib
+ root	/root
+ tmp	/tmp
+ usr	/usr
+ var	/var

- Aktuális könyvtár
 - Ebben dolgozunk. Bármely kiadott parancs itt hajtódik végre, a relatív útvonalak innen indulnak. (TIK (N46.246970, E20.142445))
- Relatív útvonal
 - Egy fájl vagy könyvtár nevének megadása az aktuális könyvtárhoz képest. (Dóm (+0.001992, +0.006720))
 - gyakorlat, .., ../h531674
- Abszolút útvonal
 - Egy fájl vagy könyvtár nevének megadása a gyökérkönyvtárból kiindulva. (Dóm (N46.248962, E20.149165))
 - /home/gertom/gyakorlat, /home, /home/h531674

Aktuális könyvtár abszolút útvonala	Relatív útvonal	Abszolút útvonal
/home/gertom	gyakorlat	/home/gertom/gyakorlat
/home/gertom	..	/home
/home/gertom	../h531674	/home/h531674
/bin	bash	/bin/bash
/bin	..	/
/bin	../etc	/etc

- Speciális jelentésű fájlnevek
 - / A könyvtárhierarchia gyökere.
 - .. A hierarchiában egy szinttel feljebb lévő könyvtár.
 - . Az aktuális könyvtár.
- A .-tal kezdődő nevek rejtett fájlt vagy könyvtárat jeleznek. Ezek bizonyos műveletek esetén rejtve maradnak, azaz nem veszünk róluk tudomást.



- A kabinetes gépeken egy `valaki` nevű felhasználó létezik, aki jelszó nélkül tud bejelentkezni. A home könyvtára a lokális gépen létező `/home.local/valaki`.
- A `mounthome` szkript futtatása után csatolódik fel a központi szerveren lévő saját hallgatói home könyvtár a `/home/hxxxxxx` pontra. Ez az `umounthome` szkript futtatásáig, vagy a rendszer újraindításáig elérhető marad!



- 1** **Bemutakozás**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2** **Linux**
 - Alapfogalmak
 - **Linux parancsok**
 - Linux shell
 - Felhasználók
 - Hálózat
- 3** **Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszoátlag adott elemszámmra
 - Csúszoátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4** **Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai

- 5** **Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.

- 6** **Folyamatábra és struktúradiagram**
- 7** **Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- Sztringek
- Pointerek és tömbök C-ben
- Rekord adattípus
- Függvény pointer
- Halmaz adattípus
- Flexibilis tömbök
- Láncolt listák
- Típusokról C-ben

- 8** **10**
 - Alapok
 - Adatállományok

- 9** **C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok

- 10** **Gyakorlati kérdések**
 - Memóriahasználát
 - Gyakori C hibák
 - where.c felboncolva

Linux parancsok

„De miért?”

- Linux-on ez a „természetes”
- A parancsértelmező nyelve egy olyan programozási nyelvnek is felfogható, aminek a linux parancsok az utasításai
- Távolról, grafikus felület nélkül is használható
- Tudni kell parancssorból programot fordítani, amihez nem árt, ha valaki parancssorból tudja kezelni a fájlrendszert
- Néha egyszerűbb és gyorsabb egy paranccsal elintézni valamit, mint végigkattintgatni a grafikus felület húsz elemén



Linux parancsok

Általános alak

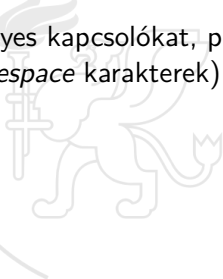
- Egy parancs általános alakja:
 - parancs kapcsolók paraméterek

parancs A parancs (vagy program) neve.

kapcsolók Általában kötőjellel kezdődő paraméter, ami a parancs/program működését befolyásolja.

paraméterek A parancs paraméterei, adatok, amiket fel fog dolgozni.

- Az egyes kapcsolókat, paramétereket szóközzel vagy tabulátorral (*whitespace* karakterek) választjuk el egymástól.



Linux parancsok

A fájlrendszer műveletei

pwd **print working directory** – az aktuális könyvtár lekérdezése

ls **list** – könyvtár tartalmának kilistázása

cd **change directory** – az aktuális könyvtár megváltoztatása

mkdir **make directory** – könyvtár létrehozása

rmdir **remove directory** – könyvtár törlése

mv **move** – fájl/könyvtár mozgatása/átnevezése

cp **copy** – fájl (vagy könyvtár) másolása

rm **remove** – fájl (vagy könyvtár) törlése

ln **link** – fájl (vagy könyvtár) linkelése

chmod **change mode** – fájl (vagy könyvtár) hozzáférési jogainak megváltoztatása

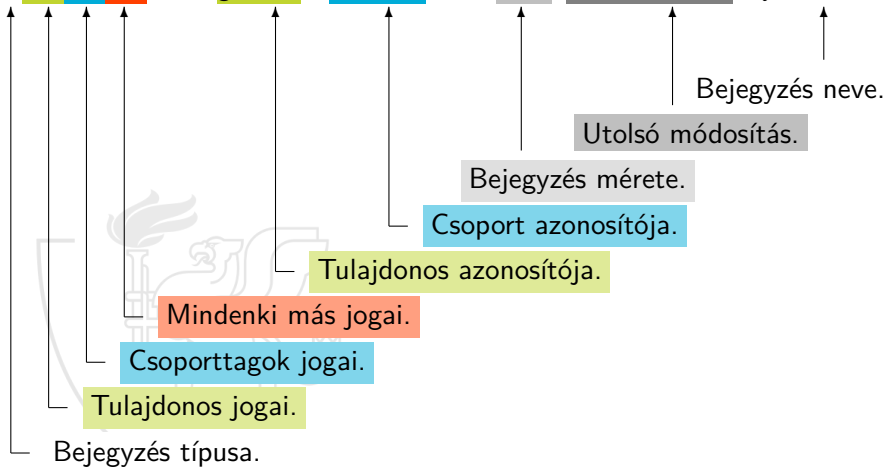
- A linux többfelhasználós rendszer. A fájlokhoz hozzáférést valamilyen módon korlátozni kell, ehhez rendelkezésre áll:
 - 3 osztály
 - Tulajdonos (User), Csoport (Group), Többiek (Others)
 - 3 féle jog
 - Olvasás (Read), Írás (Write), Végrehajtás (eXecute)
- Minden egyes fájlhoz el van tárolva némi plusz információ:
 - Tulajdonos
 - Minden egyes fájl vagy könyvtár pontosan egy felhasználóhoz tartozik.
 - Csoport
 - Minden egyes fájl vagy könyvtár pontosan egy csoportba tartozik.
 - Minden egyes felhasználó több csoportba tartozhat.
 - Utolsó módosítás dátuma
 - A fájlhoz tartozó adat helye (inode)
 - ...

Jogosultságok

```
$ ls -ld out.txt Gyakorlat
```

```
-rw-rw---- 1 gertom inf2000  
drwxr-xr-x 5 gertom inf2000
```

```
19 May 7 12:03 out.txt  
1024 Feb 7 2004 Gyakorlat
```



Linux parancsok

Szövegfájlok kezelése

- `cat` – *fájlok teljes megjelenítése*
- `more` – *fájlok megjelenítése oldalanként*
- `less` – *fájlok megjelenítése oldalanként*



Linux parancsok

Egyéb parancsok

`man` – *manual-ok, leírások*

`info` – *manual-ok, leírások*

`passwd` – *jelszóváltás* **(A kabinetben nem ezt kell használni, hanem a <http://www.inf.u-szeged.hu/jelszo> oldalról elérhető jelszóváltó űrlapot!)**

`exit` – *kilépés*



- 1 **Bemutakozás**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2 **Linux**
 - Alapfogalmak
 - Linux parancsok
 - **Linux shell**
 - Felhasználók
 - Hálózat
- 3 **Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszóátlag adott elemszámmra
 - Csúszóátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4 **Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai

- 5 **Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.

- 6 **Folyamatábra és struktúradiagram**
- 7 **Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- Sztringek
- Pointerek és tömbök C-ben
- Rekord adattípus
- Függvény pointer
- Halmaz adattípus
- Flexibilis tömbök
- Láncolt listák
- Típusokról C-ben

- 8 **10**
 - Alapok
 - Adatállományok

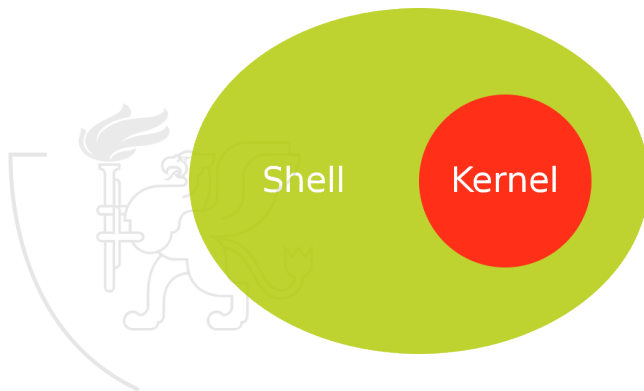
- 9 **C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok

- 10 **Gyakorlati kérdések**
 - Memóriahasználat
 - Gyakori C hibák
 - where.c felboncolva

Shell

Parancsértelmező

- A shell is „csak” egy program, de ez teszi lehetővé, hogy az operációs rendszer magjával, a kernellel kommunikálni tudjunk.
- Egy gyakori ábra:



- Sokféle van belőle
 - sh, bash, csh, dash, ksh, zsh
- Főbb funkciói
 - Kapcsolattartás a kernel és a felhasználó között.
 - Mintaillesztés.
 - Programok indítása.
 - Környezeti változók kezelése.
 - Be- és kimenet kezelés.
- Az eddig látott parancsokat is a shell értelmezi.
- A Shell egy úgynevezett prompt-tal jelzi, ha kész a parancsaink fogadására.
 - Ez általában tartalmazza a felhasználói azonosítót, a gép nevét, az aktuális könyvtárat, de egyéni ízlés szerint megváltoztatható.
 - A példáinkban a promptot a sor eleji \$ jel jelöli.

- A shell a megadott mintát mindig fájlokra és könyvtárakra próbálja illeszteni.
- Paraméterek megadásánál használhatunk joker (wildcard) karaktereket.

? Pontosán egy tetszőleges karaktert helyettesít.

* Akárhány tetszőleges karaktert helyettesít.

[...] A [és] jelek között felsorolt karaktereket helyettesítheti.

\c A c speciális karaktert jelenti, ahol c lehet:

- \, szóköz, ', ", ?, *, [,], ' ,



- " (idézőjel) Néhány behelyettesítés működik.

```
$ echo "x\"ab\"y\"  
x"aby"  
$ echo "$HOME"  
/home/gertom
```

- ' (apoztróf) Semmit sem helyettesít, karakterről karakterre másol.

```
$ echo 'x\'ab\'y\'  
x\aby\  
$ echo '$HOME'  
$HOME
```



- Ha egy szövegfájlokkal dolgozó linux programot úgy indítunk el, hogy nem adunk meg neki fájlnevet, akkor az általában automatikusan a *standard input*-ját (bemenet) fogja használni, azaz alapesetben nekünk kell ott helyben begépelnünk a feldolgozandó szöveget.
 - Ez a gépelés a `<ctrl>+d` billentyűkombináció lenyomásáig tart (amit `^d`-vel szokás jelölni).
 - A `^d`-vel vigyázzunk, mert a shell is egy ilyen program!
- Az ilyen programok általában a *standard output*-ra (kimenet) írnak, azaz alapesetben a képernyőn jelenik meg az eredmény.
- Minden programnak van még egy kimenete, a *standard error* (hiba), ahová a hibaüzeneteket küldi. Alapesetben ez is a képernyőn jelenik meg (de független a standard outputtól).

- A linux egyszerre több programot is képes futtatni.
- A programok alaphelyzetben induláskor előtérben kezdenek futni, azaz a standard inputon keresztül fogják a billentyűzetet, így amíg nem végeznek, addig nem kapjuk vissza a prompt-ot.
- Az ilyen programokkal két dolgot lehet csinálni:
 - `<ctrl>+c` Azonnal *megszakítja* a program futását, a végrehajtás befejeződik. (^c)
 - `<ctrl>+z` *Leállítja* a program futását, de az később folytatható. (^z)
- Egy programot el lehet egyből a háttérben indítani, ha a parancssor végére egy & jelet teszünk.
 - Háttérben csak olyan programok futhatnak, amelyeknek a standard inputon nincs szükségük adatra, vagy ezt egy fájlból átirányítással kiküszöböltük.
 - Általában grafikus programokat, tömörítéseket vagy nagyobb fájlműveleteket szokás így indítani.

Több program indítása sorban

- Ha több programot sorban egymás után szeretnénk végrehajtatni, akkor soroljuk fel őket egyetlen parancssorban.
- Az egyes parancsok elválasztására többféle jelet használhatunk:
 - ; Egmás eredményeitől függetlenül lesznek sorban végrehajtva.
 - && A másodikat csak akkor hajtja végre, ha az első sikerült (és).
 - || A másodikat csak akkor hajtja végre, ha az első **nem** sikerült (vagy).

```
$ cp out.txt Gyakorlat && rm out.txt
$ rm out.txt ; cp afonya.tex Gyakorlat
$ mv out.txt Gyakorlat || mv in.txt Gyakorlat
```

- 1 Bemutatók**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2 Linux**
 - Alapfogalmak
 - Linux parancsok
 - Linux shell
 - **Felhasználók**
 - Hálózat
- 3 Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszóátlag adott elemszámmra
 - Csúszóátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4 Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai
- 5 Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.
- 6 Folyamatábra és struktúradiagram**
- 7 Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- 8 Sztringek**
 - Pointerek és tömbök C-ben
 - Rekord adattípus
 - Függvény pointer
 - Halmaz adattípus
 - Flexibilis tömbök
 - Láncolt listák
 - Típusokról C-ben
- 8 10 Alapok**
 - Alapok
 - Adatállományok
- 9 C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok
- 10 Gyakorlati kérdések**
 - Memóriahasználat
 - Gyakori C hibák
 - where.c felboncolva

- Mint arról már volt szó, a linux egy többfelhasználós rendszer, vagyis ugyanazt a rendszert több felhasználó is jogosult használni, akár egyidőben is.
 - A felhasználóknak a fájlrendszerre vonatkozó jogosultságaik vannak (ezeket már láttuk).
 - A felhasználók tudhatnak egymásról.



- 1 Bemutató**
 - Kurzus információk
 - A SZTE és az informatikai képzés
- 2 Linux**
 - Alapfogalmak
 - Linux parancsok
 - Linux shell
 - Felhasználók
 - **Hálózat**
- 3 Gyors C áttekintés**
 - Bevezető
 - Pénzváltás (1. verzió)
 - Pénzváltás (2. verzió)
 - Röppálya számítás
 - Röppálya szimuláció
 - Az év napja
 - Csúszoátlag adott elemszámmra
 - Csúszoátlag parancssorból
 - Basename standard inputról
 - Basename parancssorból
 - Tér legtávolabbi pontjai
 - A nappalis gyakorlat értékelése

- 4 Alapok**
 - Alapfogalmak
 - A programozás fázisai
 - Algoritmus vezérlése
 - A C nyelvű program
 - Szintaxis
 - A C nyelv elemi adattípusai
 - A C nyelv utasításai
- 5 Vezérlési szerkezetek**
 - Bevezetés
 - Szekvenciális vezérlés
 - Függvények
 - Szelekciós vezérlések
 - Ismétléses vezérlések 1.
 - Eljárásvezérlés
 - Ismétléses vezérlések 2.
- 6 Folyamatábra és struktúradiagram**
- 7 Adatszerkezetek**
 - Az adatkezelés szintjei
 - Elemi adattípusok
 - Pointer adattípus
 - Tömb adattípus

- 8 Sztringek**
 - Pointerek és tömbök C-ben
 - Rekord adattípus
 - Függvény pointer
 - Halmaz adattípus
 - Flexibilis tömbök
 - Láncolt listák
 - Típusokról C-ben
- 8 10 Alapok**
 - Alapok
 - Adatállományok
- 9 C fordítás**
 - A fordítás folyamata
 - A preprocessor
 - A C fordító
 - Assembler
 - Linker és modulok
- 10 Gyakorlati kérdések**
 - Memóriahasználát
 - Gyakori C hibák
 - where.c felboncolva

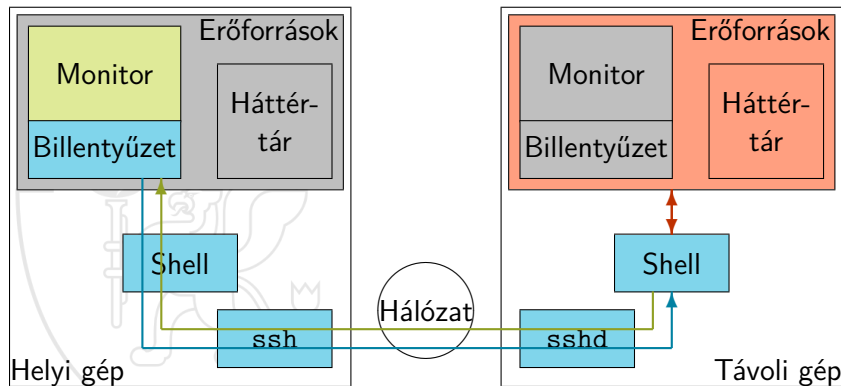
- A világ számítógépeinek nagy része az interneten keresztül TCP/IP protokollal segítségével kapcsolódik egymáshoz, amelyben egy gép azonosítása IP-cím alapján történik.
 - IPv4
 - 32 bites cím, ami 4 darab egybájtos számból áll, melyeket ponttal elválasztva decimális alakban írunk le.
 - Az SZTE gépeinek például 160.114.*.* alakú a címük.
 - Ez maximum kb. 4 milliárd cím, ami már „elfogyott”.
 - IPv6
 - 128 bites cím, ami 8 darab kétbájtos számból áll, melyeket kettősponttal elválasztva hexadecimális alakban írunk le.
 - Például: fe80:0:0:0:e2bc:4eff:fe18:9f3c.
 - Ez egy darabig talán elég lesz (a Balaton minden vízmolekulájára jutna öt ilyen cím, a Föld minden mm³-re kb. 300 millió).
- Ha valakinek 2 ilyen gépre is van azonosítója, akkor az egyikről a hálózaton keresztül elérheti a másik gépet, be tud jelentkezni rá és dolgozni tud rajta.

- Az IP-címek mellett létezik a *domain név* mint azonosító.
- Ez az emberek által könnyebben megjegyezhető formájú `www.inf.u-szeged.hu` alakú hierarchikus név.
 - Legalábbis a kétszintű hierarchia még létezik.
- A két azonosító közötti megfeleltetést a Domain Name Server-ek biztosítják, és a `host` programmal tudjuk ellenőrizni, hogy mely névhez milyen cím tartozik.
 - (Az időnként előforduló „DNS error” nem egy organikus számítógépre utal, hanem arra, hogy a beállított Domain Name Server-en keresztül nem sikerült megtudni a névhez tartozó IP-címet.)

- Ha be vagyunk jelentkezve egy gépre, az ssh program segítségével tudunk onnan egy másikra bejelentkezni.
- Az ssh használata:
 - `ssh <azonosító>@<gépnév>`
 - Az <azonosító> felhasználót bejelentkezteti a <gépnév> gépre.
 - Az <azonosító>@ rész csak akkor kell, ha az a helyi és távoli gépen nem egyezik meg.
 - A távoli gépre ugyanúgy jelszóval kell bejelentkeznünk. Ha ez sikerült, akkor a távoli gépen elindul egy shell.
 - A kapcsolatot az `exit` paranccsal (vagy `^d`-vel) zárhatjuk.

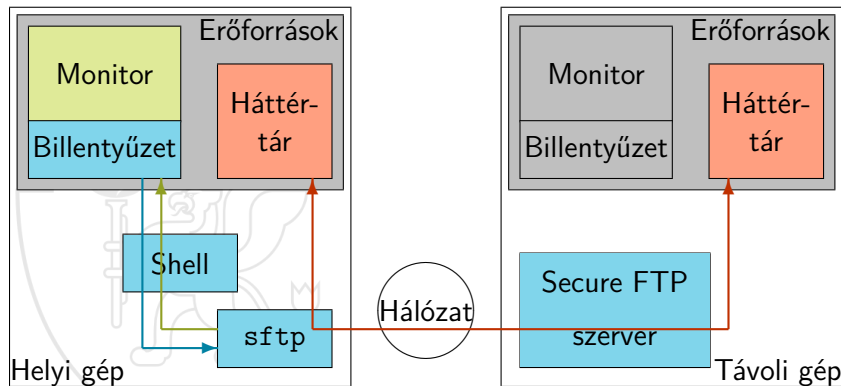


- Sikeres bejelentkezés után a távoli gépen elindul egy shell, ami a helyi géptől kapja a bemenetét, és a helyi gépre küldi a kimenetét.



- Ha be vagyunk jelentkezve egy gépre, az sftp program segítségével tudunk onnan fájlmozgatás céljából egy másikra bejelentkezni.
- Az sftp használata:
 - `sftp <azonosító>@<gépnév>`
 - Az <azonosító> felhasználót bejelentkezteti a <gépnév> gépre fájlmozgatás céljából.
 - Az <azonosító>@ rész csak akkor kell, ha az a helyi és távoli gépen nem egyezik meg.
 - A távoli gépre ugyanúgy jelszóval kell bejelentkeznünk. Ha ez sikerült, akkor létrejön a kapcsolat a távoli géppel, amin keresztül a megfelelő parancsokkal fájlokat mozgathatunk a két gép között.
 - A kapcsolatot az `exit` vagy `bye` paranccsal zárhatjuk.

- Sikeres bejelentkezés után létrejön a kapcsolat a távoli géppel, amin keresztül fájlokat mozgathatunk a két gép között.



- `scp <kapcsolók> <források> <cél>`
 - Úgy működik, mint a `cp` parancs, csak a forrásként és/vagy célként egy távoli gép könyvtára is megadható. Az útvonalakat ki lehet egészíteni a `<gépnév>`: vagy `<azonosító>@<gépnév>`: előtaggal.
 - Ez relatív útvonal esetén a távoli gépen lévő home könyvtárat jelenti.

```
$ scp Gyakorlat/in.txt gertom@linux.inf.u-szeged.hu:  
$ scp gertom@linux.inf.u-szeged.hu:in.txt vissza.txt
```

