

# Operációs rendszerek alapjai

8. gyakorlat

# Mai gyakorlat

- Környezeti változók
- Feladatok ütemezése
- Egyéb fontos parancsok
- Feladatokhoz szükséges: [G08.tar.gz](http://G08.tar.gz)

# export

- Exportálhatunk egy változót, amit megkapnak a child process-k is
  - a child process szemszögéből egy környezeti változó lesz
    - tehát módosíthatja, viszont a parent process az eredeti értéket fogja használni
    - olyan ez, mint az érték szerinti paraméterátadás
- Létrehozásra is alkalmas, de már létező változót is exportálhatunk
  - `export var=5`
  - `var=5`  
`export var`
- Kapcsolók
  - `-f`: függvény exportálása
  - `-p`: exportált objektumok listázása

# G08F01

- Próbáld ki az `export` kulcsszó működését!
  - hozz létre egy script-t, ami kiírja egy adott (egyelőre nem létező) változó értékét
  - futtasd le a script-t és figyelj meg, hogy üres sort ír ki akkor is, ha terminálban definiálsz a változót
  - exportáld a változót és futtasd újra a script-t
  - változtasd meg a változó értékét a script-ben és figyelj meg, hogy a terminálban nem változik

# alias

- Megadhatunk egy másik nevet adott parancsnak
- Hosszú, sok flag-t igénylő, gyakran használt parancsok esetén érdemes használni
- `alias ll="ls -l"`
- Személy szerint nem ajánlom a gyakori használatát, mert elfelejted az alias mögötti jelentést és ha esetleg nem pont úgy kell meghívni a parancsot, akkor újra meg kell nézned a elfedett parancsot és kapcsolókat
- Script-be nem adódik át, nem lehet exportálni sem
  - csökkentené a script hordozhatóságát

# dotfiles

- Konfigurációs fájlok, beállításokat tartalmaznak egyes programokkal kapcsolatban
  - vim-hez `.vimrc`
  - bash-hez `.bashrc`
- Az elnevezés arra utal, hogy ezek gyakran rejtett fájlok
- Dotfile-ok használata növeli a rendszered hordozhatóságát
  - nekem például van egy script-em, ami feltölti az összes fontos dotfile-t a felhőbe, meg egy script, ami letölti ezeket
  - újratelepítése előtt feltöltés, utána letöltés, így gyorsan és fájdalommentesen frissítettem
- Ezen felül ezek a fájlok teljesen disztribúció függetlenek

# .bashrc

- Általában szokott lenni egy előre létrehozott .bashrc sok hasznos beállítással
  - ezeket a beállításokat kezdőként nem érdemes megváltoztatni
- Lehet függvényeket is definiálni, amiket ezentúl tudsz használni a terminálban
- Lehet változókat és függvényeket exportálni
  - például \$PATH változtatásait így tudod maradandóvá tenni
- Lehet létrehozni alias-okat
- És még sok más

# exec

- Lefuttat egy programot subshell nélkül
  - egész pontosan a program helyettesíteni fogja az aktuális shell-t
- Subshell létrehozásának mellőzése természetesen kíméli az erőforrásokat
- Akkor is hasznos, hogyha nem kell visszaadni a vezérlést a parent process-nek
  - például script-ben, ha tudod, hogy már nem akarsz visszatérni a script-be

# G08F02

- Tegyük fel, hogy bash az alapértelmezett shell egy rendszeren (például cortex), de inkább sh-t szeretnél használni
- Vizsgáld meg, hogy mi történik, ha egyszerűen lefuttatod az sh parancsot!
- Vizsgáld meg, hogy mi történik, ha exec-el futtatsz!

# date

- Print or set the system date and time
- Script-ben szokás használni például log vagy backup készítéshez stb.
- Rengeteg formátum elérhető
- `date +FORMAT`

# cron

- Ütemezésre alkalmas program
- Szabályok és futtatandó parancsok segítségével lehet irányítani a daemon-t
- `crontab` paranccsal nyithatjuk meg a feladatokat tartalmazó fájlt
  - `-l` kapcsolóval kiírhatjuk a tartalmát
  - `-e` kapcsolóval szerkeszthetjük
    - első alkalommal választani kell, hogy milyen szerkesztővel nyissa meg
  - `-r` kapcsolóval törölhetjük
  - `-u` kapcsolóval megadhatjuk, hogy melyik felhasználó feladataival foglalkozunk
    - ezt csak superuser teheti meg
  - lehet superuser szintű parancsokat is végrehajtani, ezeket a `sudo crontab` paranccsal érthetjük el
- Hibákról vagy logokról tud e-mailt küldeni a felhasználóhoz rendelt vagy a MAILTO változóban lévő címre (egyetemi szerverekről hallgatók nem küldhetnek e-mailt)

# crontab

- Üres és #-el kezdődő sorokat nem veszi figyelembe
- Szintaxis:
  - öt paraméter vagy speciális trigger
    - perc (0-59)
    - óra (0-23)
    - nap (1-31)
    - hónap (1-12 vagy három betűs angol rövidítés)
    - hét napja (0-6 vagy három betűs angol rövidítés)
    - speciális triggerok például: @reboot, @yearly, @monthly, @weekly, @daily, @hourly
    - felsorolás vesszővel, intervallum kötőjellel, minden érték wildcard-al, lépésköz /-el
  - parancs
- Cron nem szokásos bash-t hív meg, sok funkció (pl command substitution) nem elérhető

# crontab példák

- minden percben
  - \* \* \* \* \*
- minden páros percben
  - \*/2 \* \* \* \*
- minden páratlan percben
  - 1-59/2 \* \* \* \*
- minden nap éjfélkor
  - 0 0 \* \* \*
- minden nap reggel és este nyolckor
  - 0 8,20 \* \* \*

# crontab példák

- minden hétköznapi éjfélikor
  - 0 0 \* \* 1-5
- minden hétvégén éjfélikor
  - 0 0 \* \* 6,0
- minden hónap első és tizenötödik napján és szerdánként éjfélikor
  - 0 0 1,15 \* 3
- minden hónap első napján 14:15-kor
  - 15 14 1 \* \*
- minden héten hétfőtől péntekig 22:00-kor
  - 0 22 \* \* 1-5

# G08F03

- Írj egy cron job-t, ami minden percben megnyit egy böngészőt a kedvenc weboldaladdal!
  - `* * * * * export DISPLAY=:0 && /bin/firefox --new-window google.com`

# G08F04

- Írj egy cron job-t, ami minden percben megnyit egy terminált és kiírja, hogy Hello World! A terminál maradjon nyitva öt másodpercig, majd zárjon be!
  - `* * * * * export DISPLAY=:0 && /bin/gnome-terminal -- /bin/sh -c 'echo Hello World!; sleep 5'`
- Módosítsd úgy a cron job-t, hogy ne zárjon be a terminál!
  - `* * * * * export DISPLAY=:0 && /bin/gnome-terminal -- /bin/sh -c 'echo Hello World!; sleep 5; exec bash'`

# G08F05

- Írj egy Hello World script-t a dátum kiírásának kiegészítésével!
- Írj egy cron job-t, ami minden percben meghívja ezt a script-t és átirányítja egy fájlba!
  - `* * * * * ~/hello.sh >> file`

# time

- Run programs and summarize system resource usage
- Kimenetben kiírja az eltelt valós időt, user időt és system időt
  - valós idő a program futásának elejétől a végéig eltelt idő
  - user idő az, amit a CPU user módban töltött, például tömb iterálása
  - system idő az, amit a CPU system módban töltött, például exec hívása
- Formátum átállítható

# xargs

- Build and execute command lines from standard input
- Standard inputról (tehát pipe-ből is) olvas argumentumokat és végrehajt adott parancsot
- `inputs | xargs command`
  - inputokra végrehajta a `command` parancsot, a `command inputs` parancshoz hasonlóan
  - lehet szabályozni, hogy hány soronként hajtsa végre a parancsot
    - például ha létrehozni kívánt mappák neveit adom be, mindegy, hogy egyesével vagy együtt

```
mkdir dir1
mkdir dir2 == mkdir dir1 dir2 dir3
mkdir dir3
```

# xargs

- Fontosabb kapcsolók
  - `-a filename`: adott fájlból olvassa az argumentumokat
  - `-L num`: num soronként olvassa be a bemenetet
  - `-I replace-str`: paraméterek létrehozása, az argumentumot ide helyettesíti be
    - ekkor mindenképpen soronként olvas be, tehát `-L 1` kapcsoló be lesz kapcsolva
  - `-p`: interaktív mód, minden parancs lefuttatása előtt megkérdezi, hogy futtassa-e vagy sem
  - `-t`: verbose mód, minden parancs lefuttatásakor kiírja az adott parancsot

# G08F06

- Töltsd le és csomagold ki a G08.tar.gz fájlt!
  - vagy másold le a cortex-en a /home/vagmi mappából
- Hasonlítsd össze a find parancs beépített végrehajtó kapcsolójának és az xargs parancsnak a futásidejét a kicsomagolt fájlkon!
  - például egy grep végrehajtásával
- Próbáld ki többféle verzióban, például beépített végrehajtó esetén konkatenálással és anélkül, illetve az xargs parancs esetén különböző maximális sorokkal!

# HF02 - 15 pont

- One-vs-rest multiclass perceptron osztályozót kell írni awk-ban
- Cortex szerveren /home/vagmi mappában HF02.tar.gz fájlban
  - garantáltan lineárisan szeparábilis, konvergálni fog az osztályozó
  - 2D adat egyébként, de a kód ezt ne használja ki
- Cortex 9500-s portján fut egy API, HTTP status kódokkal vagy json-el válaszol
  - tehát cortex szerveren `http://localhost:9500/endpoint` címen érhető el, wget vagy curl
- Osztályozó tanítása után kérni kell legalább egy új adatot a /HF02/new\_data endpoint-ról
  - POST kérés json formátumban, aminek van "NEPTUN" mezője érvényes Neptun kóddal
- Prediktált osztályt fel kell tölteni a /HF02/check\_prediction endpoint-ra, ami ellenőrzi a választ
  - POST kérés json formátumban, "NEPTUN" mezővel, "data" mezővel és "prediction" mezővel
- Megoldást tempus szerverre a /home/vagmi/Desktop/HF02 mappába kell feltölteni
  - ha több fájl (pl bash és awk script külön), akkor tömörítve

# Érdekességek, olvasnivalók

- [crontab.guru](https://crontab.guru)
  - nagyon jó gyakorlás