

Operációs rendszerek alapjai

IX. gyakorlat

cron

Daemon

- Görög eredetű szó
 - A görög mitológiában az isteneknél alacsonyabb, az embereknél magasabb rendű lények.
 - Programozók ezt a kifejezést először az MIT-n használták (Maxwell's demon)
- Háttérben futó folyamatok
 - Nincsenek terminálhoz kötve
- Valamilyen szolgáltatást nyújtanak, vagy felügyeleti feladatokat látnak el.
- Daemon folyamatok megtekintése:
 - `ps axf`
 - Ha a TTY oszlopban kérdőjelet látunk, akkor az adott folyamat nincs terminálhoz (TeleTYpewriter) kötve, vagyis nincs interaktív felülete amelyen keresztül vezérelhetjük, akkor ez egy daemon.

cron

- Chronos az idő deifikált fogalma az ókori görögöknél.
- A Unix-szerű rendszerekben a cron egy daemon, amely más programokat indít előre meghatározott időpontokban.
- Működését kisegítő fájlok segítségével szabályozhatjuk
 - Vannak fájlok, amelyekkel azt tudjuk megadni, hogy kik futtathatják a cron-t
 - cron.allow, cron.deny
 - Paramétereket (milyen parancsokat mikor kell futtatni) tartalmazó fájlok
 - crontab
 - Minden felhasználónak külön crontab file-ja van
 - Tartalmazza azokat az utasításokat, melyek a cront vezérlik

crontab

- Ezzel a paranccsal egy-egy felhasználóra vonatkozó crontabot lehet módosítani.
 - [-u user] : felhasználó megadása
 - Csak a superuser adhat meg a magáétól különböző felhasználónevet.
 - -l : kiírja a crontab fájl tartalmát
 - -e : megnyitja a crontab fájlt szerkesztésre
 - első alkalommal ki kell választani, hogy melyik szövegszerkesztővel nyissa meg
 - -r : törli a crontab fájlt

crontab

- Szintakszis:
 - Időzítés
 - 5 paraméter:
 - perc (0-59)
 - óra (0-23)
 - nap (1-31)
 - hónap (1-12 vagy 3 betűs angol rövidítés)
 - hét napja (0-6, ahol vasárnap=0 vagy 7, vagy 3 betűs angol rövidítés)
 - Parancs
 - tetszőleges parancs megadható, melynek kimenete átirányítható a szokott módon
 - Az üres és #-tel kezdődő sorokat nem veszi figyelembe a cron

crontab

- Példák:
 - **30 11 14 11 4 echo "Hello World!" > hello.txt**
 - November 14-én 11:30-kor kiírja a "Hello World!" szöveget a hello.txt fájlba
 - a hello.txt fájl a felhasználó home könyvtárában található
 - **30 11 14 nov 4 echo "Hello World!" > hello.txt**
 - Ugyanaz mint a felette lévő
 - **40 11 14 nov 4 /bin/bash hello.sh**
 - November 14-én 11:40-kor lefuttatja a hello.sh scriptet, ami a felhasználó home könyvtárában található
 - **43 11 14 nov 4 hello.sh 2>> errs.txt**
 - November 14-én 11:43-kor lefuttatja a hello.sh scriptet, ami a felhasználó home könyvtárában található, a hibaüzenet pedig az errs.txt fájlba kerül, a hello.sh-val azonos könyvtárba.

crontab

- Az időzítés során megadhatóak:
 - intervallumok is kötőjellel
 - pl.: **0 9-17 14 11 4 echo "Hello World!" >> hello.txt**
 - 9-től 17-ig minden órában kiírja a Hello World!-t
 - több időpont felsorolva, vesszővel elválasztva:
 - pl.: **0 9,17 14 11 4 echo "Hello World!" >> hello.txt**
 - 9-kor és 17-kor kiírja a Hello World!-t
 - Adott paraméter esetén minden lehetséges számérték *-gal
 - pl.: **0 * 14 11 4 echo "Hello World!" >> hello.txt**
 - minden órában kiírja a Hello World!-t
 - Lépésköz az adott paraméter után / jellel
 - pl.: *** /5 * 14 11 4 echo "Hello World!" >> hello.txt**
 - minden 5. percben, minden órában kiírja a Hello World!-t

crontab

- A cron számára megadhatunk változókat is
 - SHELL=/bin/bash
 - Meghatározza, hogy milyen shellt használjon a parancsok értelmezéséhez
 - MAILTO=root@example.com
 - Ezzel a változóval tudjuk beállítani, hogy milyen email címre küldje a futás eredményét
 - PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
 - Az útvonalakat a PATH változóval állíthatjuk be

crontab

- Időzítés példák:
 - minden percben
 - minden órában, félkor
 - minden évben
 - minden héten
 - 30 másodpercenként
 - bootoláskor

crontab

- Időzítés példák:
 - minden percben
 - * * * * *
 - minden órában, félkor
 - 30 * * * *
 - minden évben
 - 0 0 1 1 *
 - @yearly
 - minden héten
 - 0 0 * * mon
 - @weekly
 - 30 másodpercenként
 - * * * * * sleep 30;
 - bootoláskor
 - @reboot

G09F01

- A cron segítségével egy log.txt fájlba
 - logoljuk a dátumot minden 2. percben
 - minden 3. percben pedig azt, hogy “eltelt 3 perc”

G09F02

- A cron segítségével
 - Készítsünk egy scriptet ami a felhasználó egy adott mappájában található log.txt fájlról készít biztonsági mentéseket minden órában kétszer, egészkor és félkor!

G09F03

- A cron segítségével
 - Készítsünk egy scriptet, ami fájlokat hoz létre egy adott mappában, 5 percenként, resX.txt néven, ahol X mindig eggyel nagyobb egész szám.
 - Egy másik scriptet, ami minden csütörtökönként minden órában 15, 40 és 55-kor logolja egy num_of_files.txt mappába, hogy hány fájl van az adott mappában.