

# **Operációs rendszerek BSc**

**9. Gyak.**

**2022. 04. 05.**

**Készítette:**

Nyíri Dániel Bprof

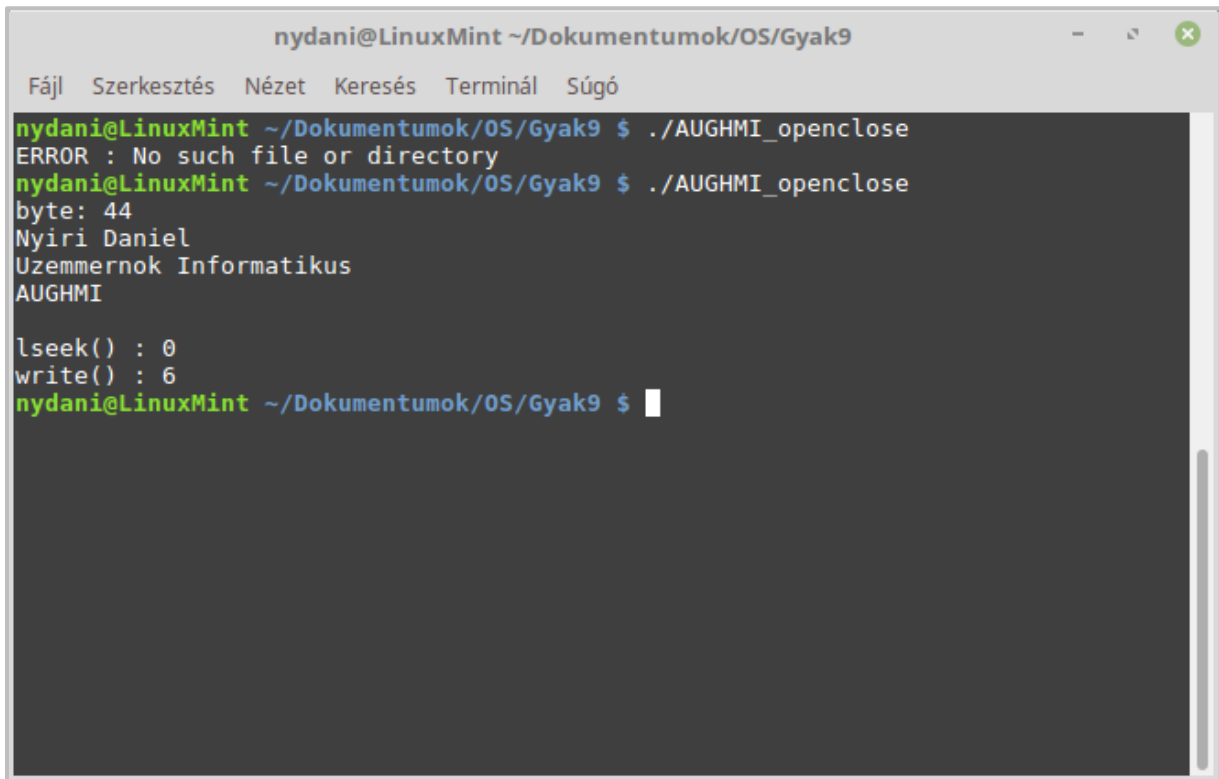
Üzemmérnök-Informatikus

AUGHMI

**Miskolc, 2022**

## 1. feladat

A tanult rendszerhívásokkal `open()`, `read()`, `write()`, `close()` fogják a rendszerhívásokat tovább hívni – írjanak egy `neptunkod_openclose.c` programot, amely megnyit egy fájlt `neptunkod.txt`, tartalma: *hallgató neve, szak, nektunkód*.

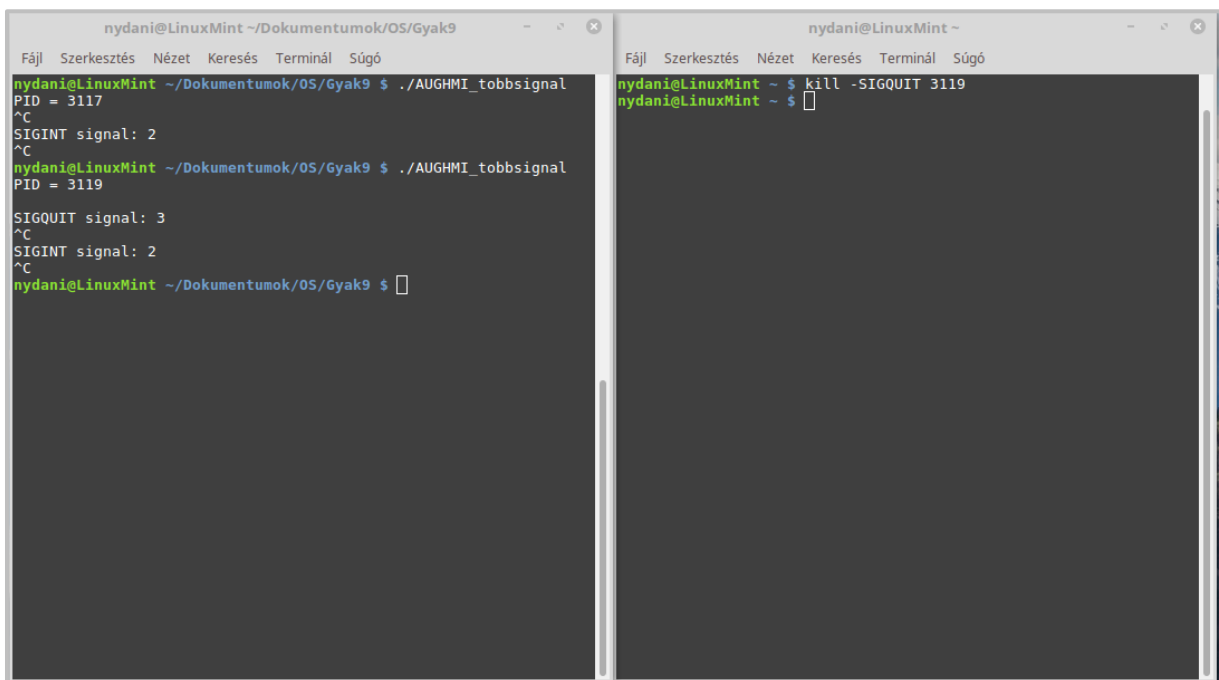


```
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9
Fájl Szerkesztés Nézet Keresés Terminál Súgó
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./AUGHMI_openclose
ERROR : No such file or directory
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./AUGHMI_openclose
byte: 44
Nyiri Daniel
Uzemmernok Informatikus
AUGHMI

lseek() : 0
write() : 6
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $
```

## 2. feladat

Készítse el a következő feladatot, melyben egy szignálkezelő több szignált is tud kezelni.



```
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9
Fájl Szerkesztés Nézet Keresés Terminál Súgó
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./AUGHMI_tobbsignal
PID = 3117
^C
SIGINT signal: 2
^C
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./AUGHMI_tobbsignal
PID = 3119
SIGQUIT signal: 3
^C
SIGINT signal: 2
^C
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $
```

```
nydani@LinuxMint ~
Fájl Szerkesztés Nézet Keresés Terminál Súgó
nydani@LinuxMint ~ $ kill -SIGQUIT 3119
nydani@LinuxMint ~ $
```

### 3. feladat

Adott a következő ütemezési feladat, amit az *FCFS*, *SJF* és *Round Robin (4ms)* ütemezési algoritmus alapján határozza meg a következő teljesítmény értékeket, metrikákat.

#### CPU kihasználtság kimaradt!

FCFS	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Körülfordulás
P1	0	24	0	24	0	24
P2	0	3	24	27	24	27
P3	2	6	27	33	25	31
P4	5	3	33	36	28	31

CPU kihasználtság:	
Körülfordulási átlag:	28,25
Átlag várakozás:	19,25
Válaszidők átlaga:	19,25

SJF	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Körülfordulás	Legrövidebb
P1	0	24	0	24	0	24	P2
P2	0	3	24	27	24	27	P4
P3	2	6	30	36	28	34	P3
P4	5	3	27	30	22	25	

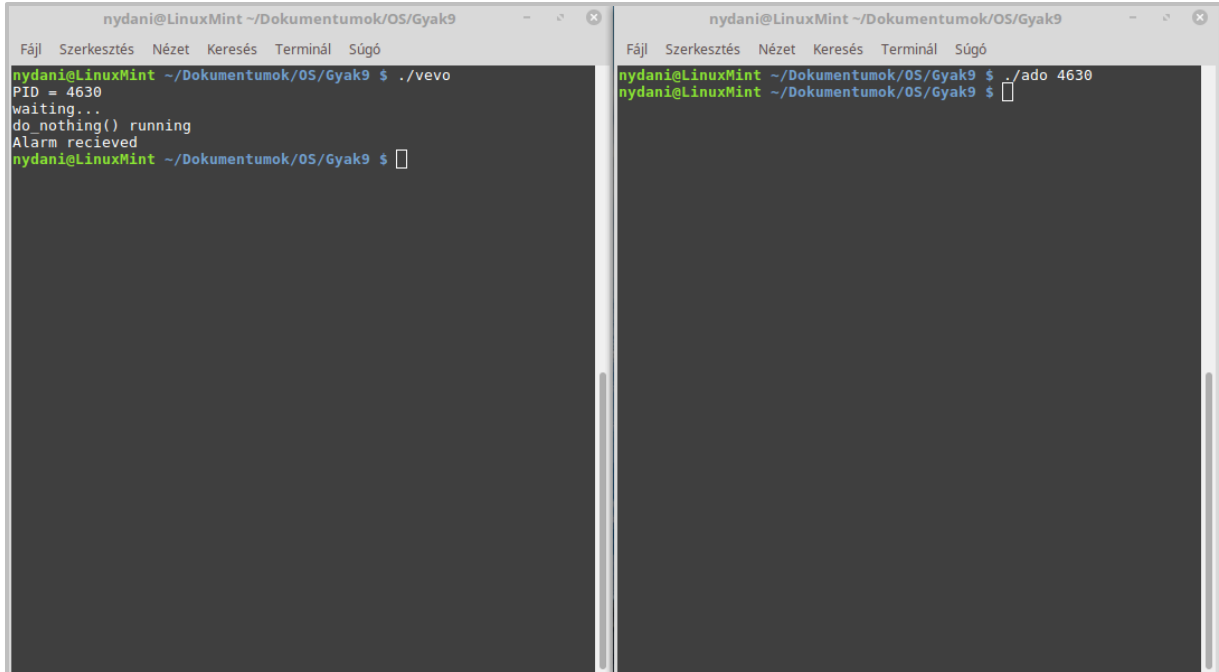
CPU kihasználtság:	
Körülfordulási átlag:	27,5
Átlag várakozás:	18,5
Válaszidők átlaga:	18,5

RR: 4ms	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Körülfordulás	Várakozó processz
P1	0, 4, 15	24, 20, 16	0, 11, 20	4, 15, 36	0, 7, 5	4, 11, 21	P2, P3, P1   P4, P3, P1
P2	0	3	4	7	4	7	P3, P1, P4
P3	2, 11	6, 2	7, 18	11, 20	5, 7	9, 9	P1, P4, P3
P4	5	3	15	18	10	13	P3, P1

CPU kihasználtság:	
Körülfordulási átlag:	18,5
Átlag várakozás:	9,5
Válaszidők átlaga:	4,75

#### 4. feladat

Írjon C programot, amelyik `kill()` segítségével `SIGALRM`-ot küld egy argumentumként megadott PID processznek, egy másik futó program a `SIGALRM`-hoz rendeljen egy függvényt, amely kiírja *neptunkódot*, továbbá `pause()` függvénnyel blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon.

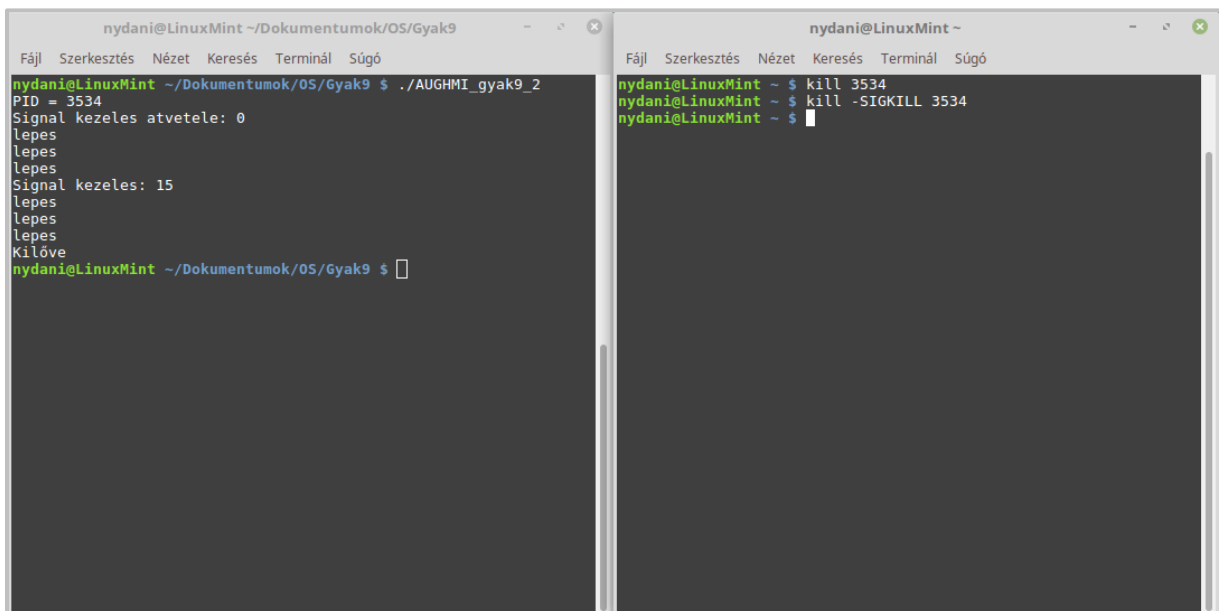


```
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./vevo
PID = 4630
waiting...
do_nothing() running
Alarm recieved
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $

nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./ado 4630
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $
```

#### 5. feladat

Írjon C nyelvű programot, amelyik a `SIGTERM`-hez hozzárendel egy függvényt, amelyik kiírja az `int` paraméter értéket, majd végtelen ciklusban fusson *3sec-ig* állandóan blokkolódva. Elindítás után egy másik shell-ben `kill` parancssal próbálja terminálni, majd `SIGKILL`-el.



```
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $ ./AUGHMI_gyak9_2
PID = 3534
Signal kezeles atvetele: 0
lepes
lepes
lepes
Signal kezeles: 15
lepes
lepes
lepes
Kilöve
nydani@LinuxMint ~/Dokumentumok/OS/Gyak9 $

nydani@LinuxMint ~ $ kill 3534
nydani@LinuxMint ~ $ kill -SIGKILL 3534
nydani@LinuxMint ~ $
```