

# **Operációs rendszerek BSc**

**9. Gyak.**

**2022. 04. 06.**

**Készítette:**

Honti Dániel BSc

Programtervező Informatikus

HR6121

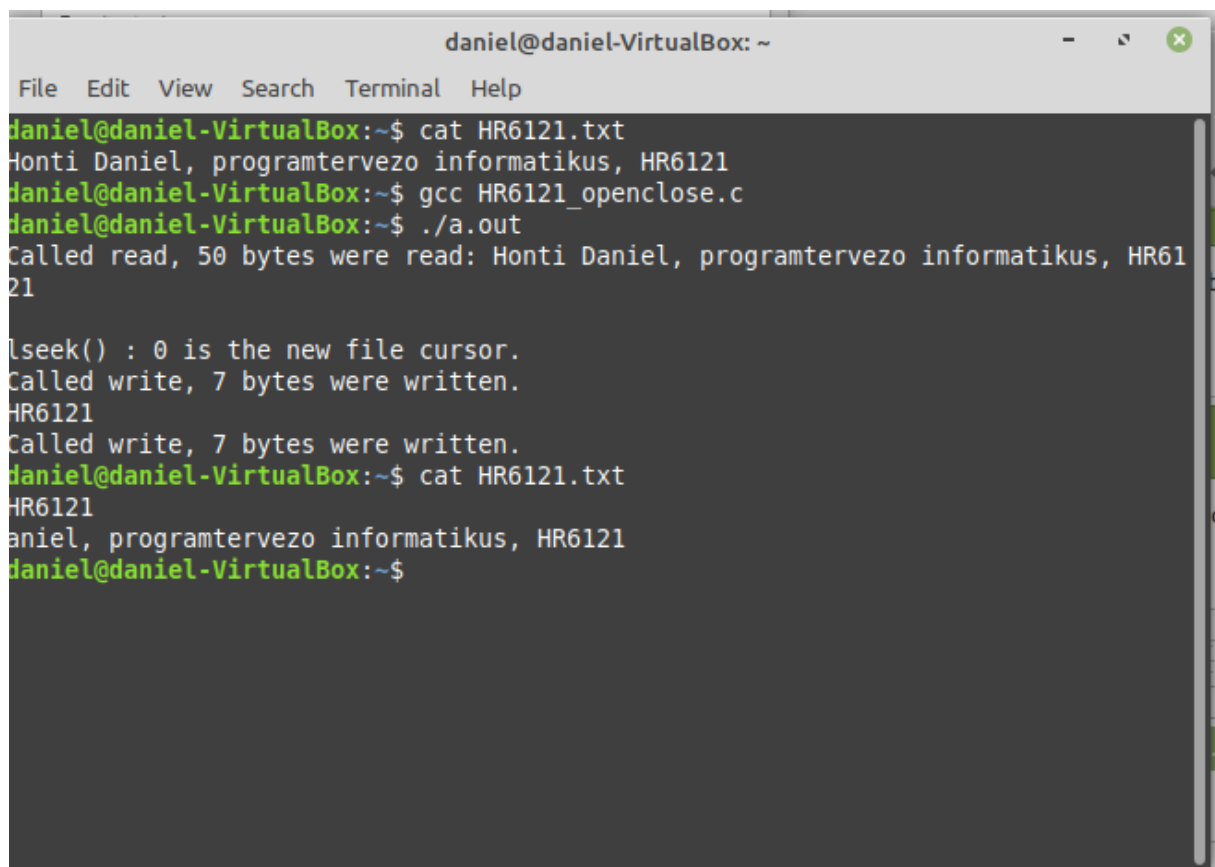
**Miskolc, 2022**

## 1. feladat

A tanult rendszerhívásokkal (open(), read()/write(), close()) - ők fogják a rendszerhívásokat tovább hívni - írjanak egy neptunkod\_openclose.c programot, amely megnyit egy fájlt – neptunkod.txt, tartalma: hallgató neve, szak , neptunkod.

A program következő műveleteket végezze:

- olvassa be a neptunkod.txt fájlt, melynek attribútuma: O\_RDWR
- hiba ellenőrzést,
- write() - mennyit ír ki a konzolra.
- read() - kiolvassa a neptunkod.txt tartalmát és mennyit olvasott ki (byte), és kiírja konzolra.
- lseek() – pozícionálja a fájl kurzor helyét, ez legyen a fájl eleje: SEEK\_SET, és kiírja a konzolra



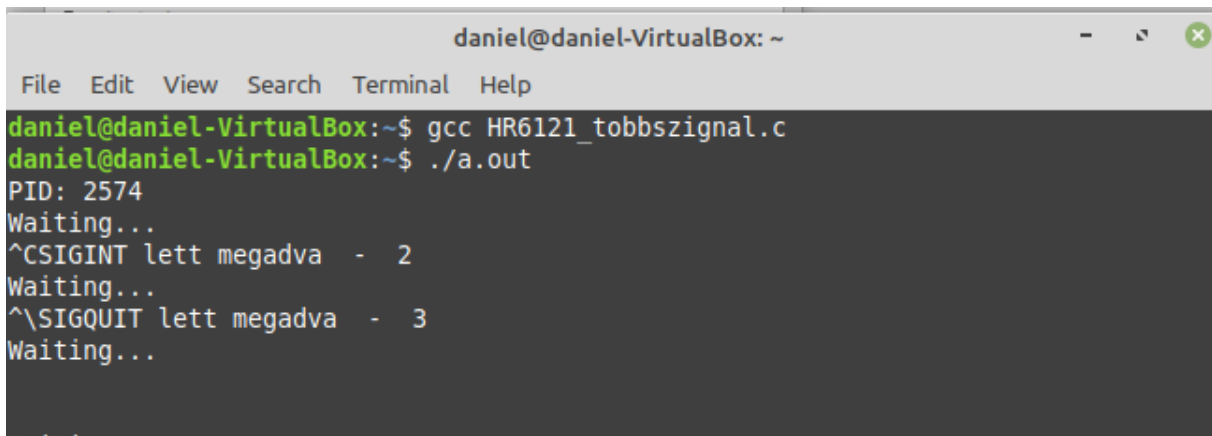
```
daniel@daniel-VirtualBox: ~  
File Edit View Search Terminal Help  
daniel@daniel-VirtualBox:~$ cat HR6121.txt  
Honti Daniel, programtervezo informatikus, HR6121  
daniel@daniel-VirtualBox:~$ gcc HR6121_openclose.c  
daniel@daniel-VirtualBox:~$ ./a.out  
Called read, 50 bytes were read: Honti Daniel, programtervezo informatikus, HR61  
21  
  
lseek() : 0 is the new file cursor.  
Called write, 7 bytes were written.  
HR6121  
Called write, 7 bytes were written.  
daniel@daniel-VirtualBox:~$ cat HR6121.txt  
HR6121  
aniel, programtervezo informatikus, HR6121  
daniel@daniel-VirtualBox:~$
```

## 2. feladat

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

- Készítsen egy szignál kezelőt (handleSignals), amely a SIGINT (CTRL + C) vagy SIGQUIT (CTRL + \) jelek fogására vagy kezelésére képes.
- Ha a felhasználó SIGQUIT jelet generál (akár kill paranccsal, akár billentyűzetről a CTRL+ \) a kezelő egyszerűen kiírja az üzenetet visszatérési értékét – a konzolra.
- Ha a felhasználó először generálja a SIGINT jelet (akár kill paranccsal, akár billentyűzetről a CTRL + C), akkor a jelet úgy módosítja, hogy a következő alkalommal alapértelmezett műveletet hajtson végre (a SIG\_DFL) – kiírás a konzolra.
- Ha a felhasználó másodszor generálja a SIGINT jelet, akkor végrehajt egy alapértelmezett műveletet, amely a program befejezése - kiírás a konzolra.

Mentés: neptunkod\_tobbsignal.c



```
daniel@daniel-VirtualBox: ~  
File Edit View Search Terminal Help  
daniel@daniel-VirtualBox:~$ gcc HR6121_tobbsignal.c  
daniel@daniel-VirtualBox:~$ ./a.out  
PID: 2574  
Waiting...  
^CSIGINT lett megadva - 2  
Waiting...  
^\\SIGQUIT lett megadva - 3  
Waiting...
```

### 3. feladat

Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR:4ms) ütemezési algoritmus alapján határozza meg következő teljesítmény értékeket, metrikákat (külön-külön táblázatba):

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

SJF	P1	P2	P3	P4
Érkezés	0	0	2	5
CPU Idő	24	3	6	3
Indulás	0	0	2	5
Befejezés	36	3	9	12
Várakozás	12	0	1	4
Körülfordulási idő	36	3	7	7

RR	P1	P2	P3	P4
Érkezés	0, 4, 15	0	2, 11	5
CPU Idő	24, 20, 16	3	6, 2	3
Indulás	0, 11, 20	4	7, 18	15
Befejezés	4, 15, 36	7	11, 20	18
Várakozás	0, 7, 5	4	5, 7	10
Körülfordulási idő	36	7	18	13