

# Operációs rendszerek BSc

7.Gyak

2022.04.05.

**Készítette:**

Tucsa Eszter Boglárka  
Mérnökinformatikus BSc  
G2QWPO

## 1.feladat:

1. 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()` – pozicionálja a fájl kurzor helyét, ez legyen a fájl eleje: `SEEK_SET`, és kiírja a konzolra.

## 2.feladat

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

- a.) Készítsen egy szignál kezelőt (`handleSignals`), amely a `SIGINT` (`CTRL + C`) vagy `SIGQUIT` (`CTRL + \`) jelek fogására vagy kezelésére képes.
- b.) Ha a felhasználó `SIGQUIT` jelet generál (akár `kill` paranccsal, akár billentyűzetről a `CTRL + \`) a kezelő egyszerűen kiírja az üzenetet visszatérési értékét – a konzolra.
- c.) 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.

d.) Ha a felhasználó másodszor generálja a `SIGINT` jelet, akkor végrehajt egy alapértelmezett műveletet, amely a program befejezése - kiírás a konzolra.

Mentés: `neptunkod_tobbszinal.c`

1. és 2. feladat Windows alatt nem működik, Linuxon jó volt, legalábbis jelentős részt. Ezek mind `.cs` file formában vannak csak így.

1	FCFS	P1	P2	P3	P4
2	Érkezés	0	0	2	5
3	CPU idő	24	3	6	3
4	Indulás	0	24	27	33
5	Befejezés	24	27	33	36
6	Várakozás	0	24	25	28
7	Körfordulás	24	27	31	31
8					
9					
10					
11	SJF	P1	P2	P3	P4
12	Érkezés	0	0	2	5
13	CPU idő	24	3	6	3
14	Indulás	12	0	3	9
15	Befejezés	36	3	9	12
16	Várakozás	12	0	1	4
17	Körfordulás	36	3	7	7
18					
19					
20	RR: 4ms	P1	P2	P3	P4
21	Érkezés	0, 4, 18, 24, 28, 32	0	2, 11	5
22	CPU idő	24, 20, 16, 12, 8, 4	3	6, 2	3
23	Indulás	0, 14, 20, 24, 28, 32	4	7, 18	11
24	Befejezés	4, 18, 24, 28, 32, 36	7	11, 20	14
25	Várakozás	0, 10, 2, 0, 0, 0	4	5, 7	6
26	Körfordulás	36	7	20	9
27					