

Operációs rendszerek BSc

9.Gyak

2022. 04. 03

Készítette:
Kocsis Zoltán
Mérnökinformatikus
Z501Ck

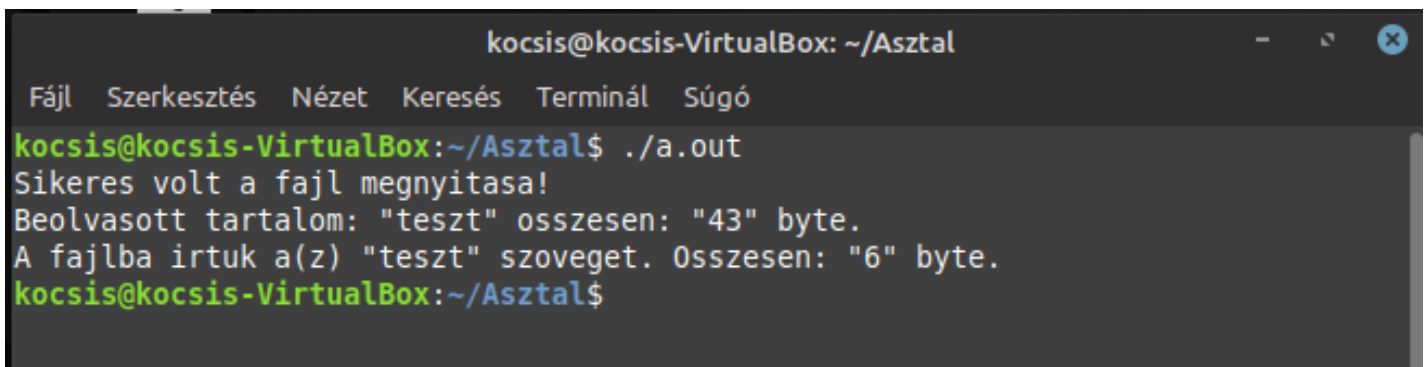
Miskolc, 2022

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:



```
kocsis@kocsis-VirtualBox: ~/Asztal
Fájl Szerkesztés Nézet Keresés Terminál Súgó
kocsis@kocsis-VirtualBox:~/Asztal$ ./a.out
Sikeres volt a fájl megnyitása!
Beolvasott tartalom: "teszt" összesen: "43" byte.
A fájlba írtuk a(z) "teszt" szöveget. Összesen: "6" byte.
kocsis@kocsis-VirtualBox:~/Asztal$
```

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

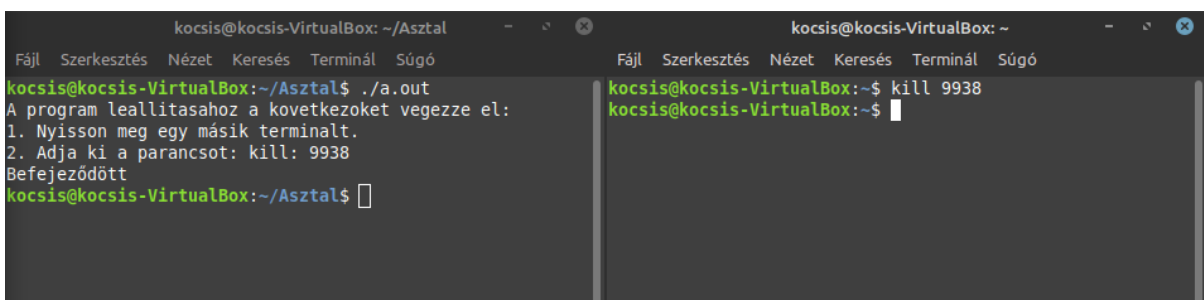
billentyűzetről a CTRL + C), akkor a jelet úgy módosítja, hogy a következő alkalommal

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_tobbsignal.c



```
kocsis@kocsis-VirtualBox: ~/Asztal
Fájl Szerkesztés Nézet Keresés Terminál Súgó
kocsis@kocsis-VirtualBox:~/Asztal$ ./a.out
A program leállításához a következőket vegezze el:
1. Nyisson meg egy másik terminált.
2. Adja ki a parancsot: kill: 9938
Befejeződött
kocsis@kocsis-VirtualBox:~/Asztal$

kocsis@kocsis-VirtualBox: ~
Fájl Szerkesztés Nézet Keresés Terminál Súgó
kocsis@kocsis-VirtualBox:~$ kill 9938
kocsis@kocsis-VirtualBox:~$
```

3. Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR: 4 ms)ütemezési algoritmus alapján határozza meg következő teljesítmény értékeket, metrikák

FCFS	érkezés	cpu idő	Indulás	Befejezés	Várakozás	Körülfordulási idő
p1	0	24	3	27	3	27
p2	0	3	0	3	0	3
p3	2	6	27	33	25	31
p4	5	3	33	36	28	31
cpu kihasználtság		156				
Körülfordulási idők átlaga		28,25				
Várakozási idők átlaga		19,25				
Válaszidők átlaga		91				

SJF	érkezés	cpu idő	Indulás	Befejezés	Várakozás
p1	0	24	12	36	12
p2	0	3	0	3	0
p3	2	6	3	9	1
p4	5	3	9	12	4
cpu kihasználtság		153			
Körülfordulási idők átlaga		27,5			
Várakozási idők átlaga		18,5			
Válaszidők átlaga		88			

RR 4ms	érkezés	cpu idő	Indulás	Befejezés	Várakozás
p1	0	24	0	4	0
	4	20	14	18	10
	18	16	20	36	2
p2	0	3	4	7	4
p3	2	6	7	11	5
	11	2	18	20	7
p4	5	3	11	14	6
cpu kihasználtság		156,4			
Körülfordulási idők átlaga		10			
Várakozási idők átlaga		7			
Válaszidők átlaga		95,6			