

Operációs rendszerek BSc

9. Gyak.

2022. 04. 06.

Készítette:

Palencsár Enikő Bsc

Mérnökinformatikus

YD11NL

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

```
enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL_openclose.c
enikop@enikop-VirtualBox:~/Desktop/OS$ nano YD11NL.txt
enikop@enikop-VirtualBox:~/Desktop/OS$ cat YD11NL.txt
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
Called write, returned that 43 bytes were written.
lseek() : 0 is the new file cursor.
Called read, returned that 43 bytes were read.
enikop@enikop-VirtualBox:~/Desktop/OS$ cat YD11NL.txt
Palencsar Eniko, mernokinformatikus, YD11NL
enikop@enikop-VirtualBox:~/Desktop/OS$
```

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_tobbszignal.c

```
enikop@enikop-VirtualBox:~/Desktop/OS$ kill -SIGINT 2681
enikop@enikop-VirtualBox:~/Desktop/OS$ kill -SIGQUIT 2681
enikop@enikop-VirtualBox:~/Desktop/OS$ kill -SIGINT 2681
enikop@enikop-VirtualBox:~/Desktop/OS$

21     printf("Waiting...\n");
22     pause();
23 }
24     printf("Terminating\n");
25     return 0;
26 }
27
28 void do_nothing(){ }
29
30 void handleSignals(int sig) {
    if (sig == SIGINT || sig == SIGQUIT) {
        printf("Waiting...\n");
        pause();
        if (sig == SIGINT) {
            printf("SIGINT lett megadva - 2\n");
        } else {
            printf("SIGQUIT lett megadva - 3\n");
        }
        printf("Waiting...\n");
    }
}

enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL_tobbszignal.c
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
PID: 2681
Waiting...
Waiting...
Waiting...
SIGINT lett megadva - 2
Waiting...
Waiting...
SIGQUIT lett megadva - 3
Waiting...
enikop@enikop-VirtualBox:~/Desktop/OS$
```

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	24	27	33
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	12	0	3	9
Befejezés	36	3	9	12
Várakozás	12	0	1	4
Várakozók	-	P1, P3	P1, P4	P1

Körülfordulási idő	36	3	7	7
---------------------------	----	---	---	---

Sorrend: P2, P3, P4, P1

RR:

RR: 10 ms	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

Sorrend:

Külön táblázatba számolja a teljesítmény értékeket!

CPU kihasználtság: számolni kell a cs: 0,1(ms) és sch: 0,1 (ms)

Algoritmus neve	FCFS
CPU kihasználtság	$36/36.4 = 98.9\%$ (4 db cs)
Körülfordulási idők átlaga	$(24+27+31+31)/4 = 28.25$ ms
Várakozási idők átlaga	$(0+24+25+28)/4 = 19.25$ ms
Válaszidők átlaga	$(0+24+25+28)/4 = 19.25$ ms

Algoritmus neve	SJF
CPU kihasználtság	$36/36.4 = 98.9\%$ (4 db cs)
Körülfordulási idők átlaga	$(36+3+7+7)/4 = 13.25$ ms
Várakozási idők átlaga	$(12+0+1+4)/4 = 4.25$ ms
Válaszidők átlaga	$(12+0+1+4)/4 = 4.25$ ms

Algoritmus neve	RR
CPU kihasználtság	$36/37 = 97.3\%$ (7 db cs, 3 db sch)
Körülfordulási idők átlaga	$(36+7+18+13)/4 = 18.5$ ms
Várakozási idők átlaga	$(0+7+5+4+5+7+10)/4 = 9.5$ ms
Válaszidők átlaga	$(0+4+5+10)/4 = 4.75$ ms

4. feladat.

Írjon C nyelvű programot, amelyik kill() seg.-vel SIGALRM-et küld egy argumentumként megadott PID-u processznek, egy másik futó program a SIGALRM-hez rendeljen egy fv.-t amely kiírja pl. neptunkodot, továbbá pause() fv.-el blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon.

Mentés. neptunkod_gyak9_1.c

```

enikop@enikop-VirtualBox:~/Desktop/05$ gcc YD11NL_gyak9_1blocker.c
enikop@enikop-VirtualBox:~/Desktop/05$ gcc YD11NL_gyak9_1blocker.c
enikop@enikop-VirtualBox:~/Desktop/05$ ./a.out 2348
enikop@enikop-VirtualBox:~/Desktop/05$

enikop@enikop-VirtualBox:~/Desktop/05$ ./a.out
PID: 2348
Signal kezelo atvetele: 0
Waiting...
YD11NL
Not blocked anymore. Terminating...
enikop@enikop-VirtualBox:~/Desktop/05$

```

5. feladat.

Írjon C nyelvű programot, amelyik a SIGTERM-hez hozzárendel egy fv-t., amelyik kiírja az int paraméter értéket, majd végtelen ciklusban fusson, 3 sec-ig állandóan blokkolódva elindítás után egy másik shell-ben kill paranccsal (SIGTERM) próbálja terminálni, majd SIGKILL-el.”

Mentés. neptunkod_gyak9_2.c

```

enikop@enikop-VirtualBox:~$ kill 2245
enikop@enikop-VirtualBox:~$ kill 2245
enikop@enikop-VirtualBox:~$ kill -SIGKILL 2245
enikop@enikop-VirtualBox:~$

1 signal(SIGALRM, doNothing);
2 while(1){
3     printf("Waiting...\n");
4     alarm(sec);
5     pause();
6 }
7
8 printf("Not blocked anymore. Terminating...\n");
9
10 return 0;
11
12 void signalHandler(int sig){
13     printf("The value of int argument is %d\n", sig);
14     return;
15 }

PID: 2245
Signal kezelo atvetele: 0
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
The value of int argument is 15
Waiting...
The value of int argument is 15
Waiting...
Waiting...
Waiting...
Waiting...
Killed
enikop@enikop-VirtualBox:~/Desktop/05$

```