

# Operációs rendszerek BSc

3. konzultáció gyakorlat

2021.04.16

Készítette: **Hudák Dániel**

Mérnökinformatikus MSc

Neptunkód: **MWB0JW**

### Operációs rendszerek – 3. konzultáció - gyakorlat

Linux OS - Rendszerhívások

Ütemezési algoritmusok (FCFS, SJF, RR)

IPC mechanizmusok

Töltse fel az aktuális mappába: **Neptunkod\_....**

Jegyzőkönyv neve: *neptunkod\_gyak3.pdf*

Forrás fájlok feltöltése

A futás eredményét is tartalmazza a jegyzőkönyv.

**Határidő:** a következő konzultáció időpontja.

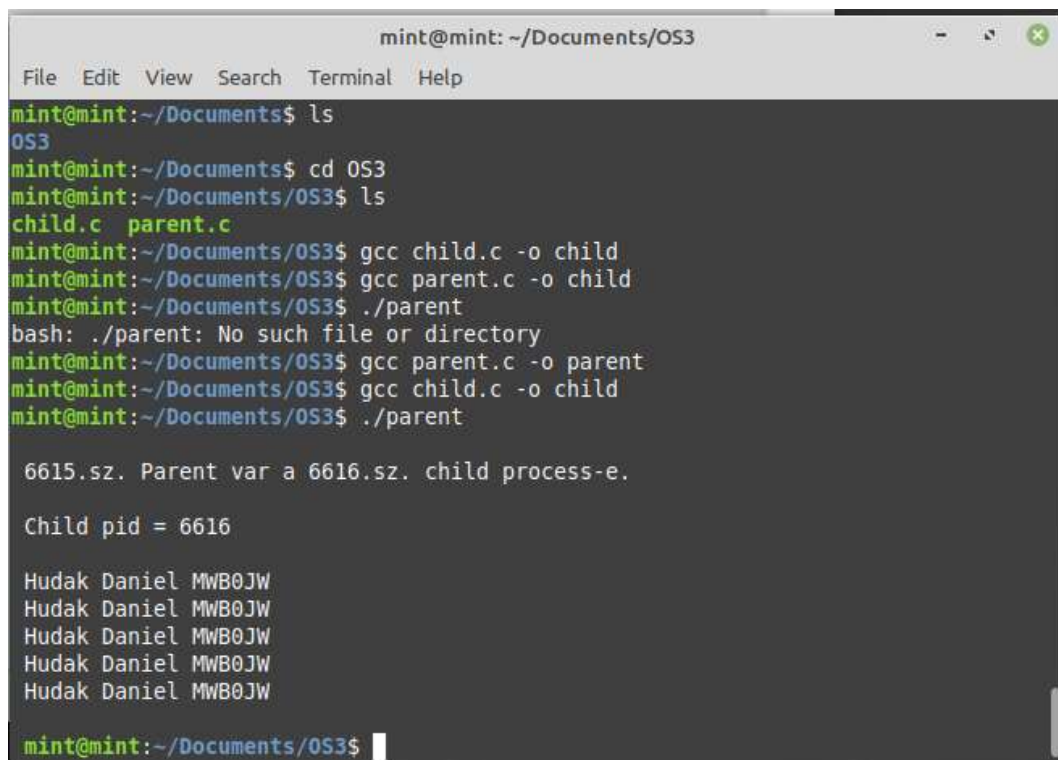
1. Készítsen egy `parent.c` és a `child.c` programokat. A `parent.c` elindít egy gyermek processzt, ami különbözik a szülőtől. A szülő megvárja a gyermek lefutását. A gyermek szöveget ír a szabványos kimenetre (5-ször) (pl. a hallgató neve és a neptunkód)!

Mentés: *parent.c*, ill. *child.c*

A `parent.c` és `child.c` programkódok megírása után a két fájlt átmásoltam a VirtualBox-on futtatott Linux Mint virtuális gépre.

A mentés helye: „`/home/mint/Documents/OS3`”.

A terminál megnyitása után az OS3 könyvtárba navigálva először gcc compiler segítségével le kell fordíttatnunk a programkódokat, majd a szülő programot futtatni.



```
mint@mint: ~/Documents/OS3
File Edit View Search Terminal Help
mint@mint:~/Documents$ ls
OS3
mint@mint:~/Documents$ cd OS3
mint@mint:~/Documents/OS3$ ls
child.c parent.c
mint@mint:~/Documents/OS3$ gcc child.c -o child
mint@mint:~/Documents/OS3$ gcc parent.c -o parent
mint@mint:~/Documents/OS3$ ./parent
bash: ./parent: No such file or directory
mint@mint:~/Documents/OS3$ gcc parent.c -o parent
mint@mint:~/Documents/OS3$ gcc child.c -o child
mint@mint:~/Documents/OS3$ ./parent

6615.sz. Parent var a 6616.sz. child process-e.

Child pid = 6616

Hudak Daniel MWB0JW
Hudak Daniel MWB0JW
Hudak Daniel MWB0JW
Hudak Daniel MWB0JW
Hudak Daniel MWB0JW

mint@mint:~/Documents/OS3$
```

2. Adott a következő terhelés esetén egy rendszer.

	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás	0	15	22	48
Befejezés				
Várakozás				

A tanult ütemezési algoritmus (FCFS, SJF, RR: 10 ms) felhasználásával határozza meg

a.) Várakozási/átlagos várakozási időt, befejezési időt?

<b>FCFS</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
<b>ÉRKEZÉS</b>	0	8	12	20
<b>CPU IDŐ</b>	15	7	26	10
<b>INDULÁS</b>	0	15	22	48
<b>BEFEJEZÉS</b>	15	22	48	58
<b>VARÁKOZÁS</b>	0	7	10	28
<b>SORREND</b>	1.	2.	3.	4.

Átlag várakozás: 11,25

<b>SJF</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
<b>ÉRKEZÉS</b>	0	8	12	20
<b>CPU IDŐ</b>	15	7	26	10
<b>INDULÁS</b>	0	15	32	22
<b>BEFEJEZÉS</b>	15	22	58	32
<b>VARÁKOZÁS</b>	0	7	20	2
<b>SORREND</b>	1.	2.	4.	3.

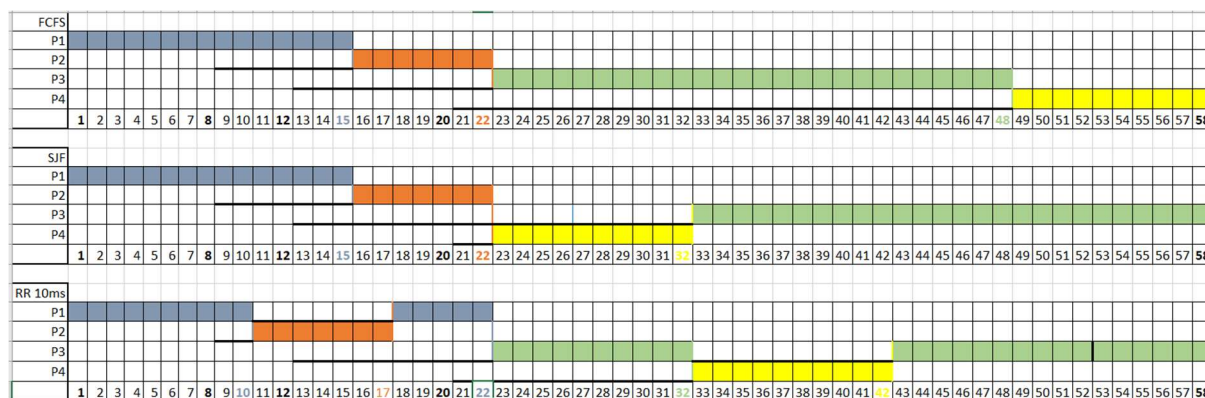
Átlag várakozás 7,25

RR 10ms	P1	P2	P3	P4
ÉRKEZÉS	0;10	8	12; 32	20
CPU IDŐ	15;5	7	26;16	10
INDULÁS	0;17	10	22;42	32
BEFEJEZÉS	10;22	17	32;58	42
VARÁKOZÁS	0;7	2	10;10	12
SORREND	1.; 3.	2.	4.;6.	5.

Átlag várakozás: 10.25

**b.)** Ábrázolja Gantt diagrammal az aktív/várakozó folyamatok futásának sorrendjét (használgon Excel or Word etc.)!

Az aktív és várakozó folyamatok futásának sorrendjét ábrázoló Gantt diagram:



**3.** Értelmezzék a mintapéldákat és oldják meg: `alarm.c`; `alarm_ado.c`;

`alarmra_var.c` - a jegyzet 68. oldalán található.

Mentés: `neptunkod_alarm.c`; `neptunkod_alarm_ado.c`;

`neptunkod_alarmra_var.c`

Signal kezelése példa: `mwb0jw_alarm.c`

Ctrl/break signal hatására először a `do_int` handler működik, majd ignorálódik.

Másodpercenként generálódik egy alarm a ciklusban, ennek a signal handler-e a `do_nothing`.

Amíg signált kap, a pause felfüggeszti a process-t.

Futtatás terminálból, a fájlt tartalmazó könyvtárból.

Először a `gcc mwb0jw_alarm.c -o mwb0jw_alarm` paranccsal lefordítatjuk a fájlt, majd a `./mwb0jw_alarm` parancsot használva lefuttatjuk.

```
mint@mint: ~/Documents/OS3
File Edit View Search Terminal Help
mint@mint:~$ pwd
/home/mint
mint@mint:~$ ls
Desktop Documents Downloads Music Pictures Public Source Templates Videos
mint@mint:~$ cd Documents/
mint@mint:~/Documents$ ls
OS3
mint@mint:~/Documents$ cd OS3/
mint@mint:~/Documents/OS3$ ls
child child.c parent parent.c
mint@mint:~/Documents/OS3$ ls
child mwb0jw_alarm.c mwb0jw_alarmra_var.c mwb0jw_named_child.c mwb0jw_unnamed.c parent.c
child.c mwb0jw_alarm_ado.c mwb0jw_named.c mwb0jw_named_parent.c parent
mint@mint:~/Documents/OS3$ gcc mwb0jw_alarm.c -o mwb0jw_alarm
mint@mint:~/Documents/OS3$ ./mwb0jw_alarm
1 varok de meddig?
2 varok de meddig?
3 varok de meddig?
4 varok de meddig?
5 varok de meddig?
6 varok de meddig?
7 varok de meddig?
mint@mint:~/Documents/OS3$
```

Signal kezelés példa: mwb0jw\_alarmra\_var.c és mwb0jw\_alarm\_ado.c

A két program párban használhatóak.

Futtatás terminál ablakból a fájlt tartalmazó könyvtárból. Először a gcc mwb0jw\_alarmra\_var.c -o mwb0jw\_alarmra\_var parancssal lefordíttatjuk a fájlt, majd ./mwb0jw\_alarmra\_var & parancsot használva futtatjuk. A terminál ablakban visszakapjuk a process pid-két (6939), ezután a fenti analógiára először lefordíttatjuk az mwb0jw\_alarm\_ado.c fájlt, majd lefuttatjuk ./mwb0jw\_alarm\_ado 6939 (megadjuk a várakozó process pid-jét).

```
mint@mint: ~/Documents/OS3
File Edit View Search Terminal Help
mint@mint:~$ pwd
/home/mint
mint@mint:~$ cd Documents/OS3/
mint@mint:~/Documents/OS3$ ls
child          mwb0jw_alarm_ado.c  mwb0jw_named_parent.c
child.c        mwb0jw_alarmra_var.c mwb0jw_unnamed.c
mwb0jw_alarm   mwb0jw_named.c      parent
mwb0jw_alarm.c mwb0jw_named_child.c parent.c
mint@mint:~/Documents/OS3$ gcc mwb0jw_alarmra_var.c -o mwb0jw_alarmra_var
mint@mint:~/Documents/OS3$ ./mwb0jw_alarmra_var &
[1] 6939
mint@mint:~/Documents/OS3$ varok de meddig?
Vegre, itt az alarm
^C
[1]+  Done                  ./mwb0jw_alarmra_var
mint@mint:~/Documents/OS3$

mint@mint:~/Documents/OS3$ ls
child          mwb0jw_alarmra_var  mwb0jw_unnamed.c
child.c        mwb0jw_alarmra_var.c parent
mwb0jw_alarm   mwb0jw_named.c      parent.c
mwb0jw_alarm.c mwb0jw_named_child.c parent.c
mint@mint:~/Documents/OS3$ gcc mwb0jw_alarm_ado.c -o mwb0jw_alarm_ado
mint@mint:~/Documents/OS3$ ./mwb0jw_alarm_ado
Segmentation fault (core dumped)
mint@mint:~/Documents/OS3$ ./mwb0jw_alarm_ado 6939
mint@mint:~/Documents/OS3$
```

4. a) Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékot, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod\_unnamed.c

Programot készítettem C nyelven (mwb0jw\_unnamed.c) a leírásnak megfelelően.

A kiírt szöveg: HD mwb0jw (karakter darabszma).

```
mint@mint:~/Documents/OS3$ gcc mwb0jw_unnamed.c -o mwb0jw_unnamed
mint@mint:~/Documents/OS3$ ./mwb0jw_unnamed
7118: szulo vagyok
7119: gyerek vagyok
7119: szoveg kuldes: HD mwb0jw (9 db. karakter)
7118: cso ellenorzes
7118: HD mwb0jw
mint@mint:~/Documents/OS3$
```

b) Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékot (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl. Keszű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod\_named.c

A kiírásnak megfelelően C nyelvű programot írtam, majd lefuttattam egyszerre. A gyerek process addig hibával tér vissza, amíg a szülő process létre nem hozza a fifo-t.



```
mint@mint: ~/Documents/OS3
File Edit View Search Terminal Help
mint@mint:~$ pwd
/home/mint
mint@mint:~$ cd Documents/OS3/
mint@mint:~/Documents/OS3$ ls
child          mwb0jw_named.c
child.c        mwb0jw_named_child
mwb0jw_alarm   mwb0jw_named_child.c
mwb0jw_alarm.c mwb0jw_named_parent
mwb0jw_alarm_ado mwb0jw_named_parent.c
mwb0jw_alarm_ado.c mwb0jw_unnamed
mwb0jw_alarmra_var mwb0jw_unnamed.c
mwb0jw_alarmra_var.c parent
mwb0jw_named   parent.c
mint@mint:~/Documents/OS3$ gcc mwb0jw_named_parent.c -o mwb0jw_named_parent
mint@mint:~/Documents/OS3$ ./mwb0jw_named_parent
fifobol kiolvasas
A hallgato neve: Hudak DanielU (29 byte)
mint@mint:~/Documents/OS3$
```

```
mint@mint: ~/Documents/OS3
File Edit View Search Terminal Help
mint@mint:~$ pwd
/home/mint
mint@mint:~$ cd Documents/OS3/
mint@mint:~/Documents/OS3$ ls
child          mwb0jw_alarmra_var  mwb0jw_unnamed
child.c        mwb0jw_alarmra_var.c mwb0jw_unnamed.c
mwb0jw_alarm   mwb0jw_named        parent
mwb0jw_alarm.c mwb0jw_named.c      parent.c
mwb0jw_alarm_ado mwb0jw_named_child.c
mwb0jw_alarm_ado.c mwb0jw_named_parent.c
mint@mint:~/Documents/OS3$ gcc mwb0jw_named_child.c -o mwb0jw_named_child
mint@mint:~/Documents/OS3$ ./mwb0jw_named_child
fifoba iras
A hallgato neve: Hudak Daniel (29 byte)
```

A futás eredményeként a kiolvasott string mindig egy U vagy egy V karakterrel több lett mint a gyermek process által beírt string.

Ha egy programként futtatom, ez a jelenség nem tapasztalható.

```
mint@mint:~/Documents/OS3$ gcc mwb0jw_named.c -o mwb0jw_named
mint@mint:~/Documents/OS3$ ./mwb0jw_named
fifoba iras
A hallgato neve: Hudak Daniel(29 byte)
kiolvasva
A hallgato neve: Hudak Daniel (29 byte)
mint@mint:~/Documents/OS3$
```