

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Palencsár Enikő**

Neptunkód: **YD11NL**

**Miskolc, 2022**

## ALGORITMUS

### A feladat leírása:

3. Adott egy számítógépes rendszer, melyben a következő

- szabad memória területek: 50k, 30k, 200k, 16k, 30k, melynek
- foglalási igénye: 20k, 30k, 10k, 100k, 60k.

Határozza meg változó méretű partíció esetén a következő algoritmusok felhasználásával: *first fit*, *worst fit* a foglalási igényeknek megfelelő helyfoglalást – táblázatos formában!

Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

### A feladat elkészítésének lépései:

Két külön táblázatban vizsgáltam a két algoritmus helyfoglalásának eredményét.

### Eredmény:

#### FIRST FIT

		Szabad partíciók				
		50	30	200	16	30
Foglalandó	20	20 + 30	30	200	16	30
	30	30 + 0	30	200	16	30
	10	0	10 + 20	200	16	30
	100	0	20	100 + 100	16	30
	60	0	20	60 + 40	16	30

Végeredmény: [20], [30], [10], 20, [100], [60], 40, 16, 30

A foglalási igények mind kielégíthetők.

A töredezettség nem lesz nagymértékű a kezdeti állapothoz képest, de azért megjelenik: keletkezik egy új 20 kB és egy 40 kB méretű partíció, a fennmaradó szabad memória összesen 106 kB, amely 4 részre oszlik.

A megmaradó partíciók közül a leghosszabb 40kB-nyi, ezért ha nagy helyigényű processz következik, az várakozni kényszerül majd.

#### WORST FIT

		Szabad partíciók				
		50	30	200	16	30
Foglalandó	20	50	30	20 + 180	16	30
	30	50	30	30 + 150	16	30
	10	50	30	10 + 140	16	30
	100	50	30	100 + 40	16	30
	60	50	30	40	16	30

Nem maradt megfelelő méretű szabad partíció, az utolsó kérelemnek egyelőre nem tudunk eleget tenni. Vagy áthelyezünk egy másik foglalást, vagy megvárjuk, amíg új partíciók szabadulnak fel.

Végeredmény: 50, 30, [20], [30], [10], [100], 40, 16, 30 és várakozik egy 60-as igény

Legnagyobb probléma: a foglalási igények nem elégíthetők ki mind, a 60 kB igényű processz várakozik. Az elején beérkező kisebb igények, melyeket valamelyik rövidebb partíció is kielégíthetne, az algoritmus miatt legfoglalják az egyetlen "hosszú" memóriarészt.

Emiatt jelen esetben mindenképpen a *first fit* algoritmus a hatékonyabb a kettő közül.

Egy másik lehetséges megoldás az eredmények javítására: BEST FIT

		Szabad partíciók				
		50	30	200	16	30
Foglalandó	20	50	20+10	200	16	30
	30	50	10	200	16	30+0
	10	50	10+0	200	16	0
	100	50	0	100+100	16	0
	60	50	0	60+40	16	0

Végeredmény: 50, [20],[10], [100], [60], 40, 16, [30]

A foglalási igények itt is mind kielégíthetők, akár csak a *first fit* esetén.

A töredezettség viszont javul a *first fit*hez képest: a 20 kB és a 30 kB hosszú partíciók helyett egyetlen 50 kB-os marad, azaz a maradék 106 kB memória itt csak 3 részre oszlik. (Az 50kB feletti helyigényű processzek itt is várakozni kényszerülnek majd.)

Ha a *best fit* és a *first fit* közül akarunk választani, mérlegelni kell, mennyivel költségesebb a *best fit* minden partíciót végignéző döntési algoritmus a viszonylag egyszerű *first fit*hez képest.

## IPC

### A feladat leírása:

15. Írjon C nyelvű programokat, ami

- létrehoz egy üzenetsort
- SIGHUP signal hatására beletesz egy üzenetet ebbe az üzenetsorba
- SIGTERM hatására szűnjön meg az üzenetsor erőforrás és lépjen is ki a program

a másik program pedig:

- SIGHUP signal hatására kiolvas ebből egy üzenetet, és kiírja a képernyőre

### A feladat elkészítésének lépései:

1. Az első program létrehozása, benne message buffer struktúra definiálása. Fájlnév: YD11NL\_1.c
2. PID kiírása a kívülről történő szignálküldéshez
3. Kulcs generálása ftok() segítségével a fájlnévből és egy számból, a kulcs segítségével üzenetsor létrehozása msgget()-tel
4. A szignálkezelő függvények beállítása és definiálása
  - a. send\_msg()-ben a buffer feltöltése, majd ftok()-kal az előbb említettel azonos kulcs legenerálása, az üzenetsor azonosítása, ezt követően pedig msgsnd()-del az üzenet üzenetsorba helyezése, ennek kiírása konzolra
  - b. delete\_msgq()-ban kulcs generálás, azonosítás, majd az üzenetsor törlése(msgctl()), ennek kiírása, és kilépés a programból
5. Végtelen ciklusban folyamatos blokkolódás pause() függvénnyel
6. A második program létrehozása, benne msg\_buffer struktúrával. Fájlnév: YD11NL\_2.c
7. PID kiírása a kívülről történő szignálküldéshez
8. A szignálkezelő függvény beállítása és definiálása
  - a. receive\_msg()-ben ftok()-kal az előző programban szereplővel azonos kulcs legenerálása után az üzenetsor azonosítása. Ezt követően ha nem létezik az üzenetsor, ezt kiírjuk, ha igen, akkor msgrcv()-vel fogadjuk az adott típusú üzenetet. Ha nincs üzenet, ezt a konzolra kiírjuk, ha van üzenet, az üzenetet írjuk ki.
9. Végtelen ciklusban folyamatos blokkolódás pause() függvénnyel
10. Futtatás, tesztelés új terminálablakból küldött szignálokkal

### A futtatás eredménye:

A két program futtatása parancssorból:

```
enikop@enikop-VirtualBox: ~/Desktop/OS/Beadando
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./a.out
PID: 2312
Az üzenetsor létrejött
Üzenet küldése: SIGHUP, üzenetsor törlése: SIGTERM, varakozás szignálra...
█

enikop@enikop-VirtualBox: ~/Desktop/OS/Beadando
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./b.out
PID: 2311
Üzenet fogadása: SIGHUP, varakozás szignálra...
█
```

Létrejön az üzenetsor a futtatással:

```
enikop@enikop-VirtualBox:~$ ipcs

----- Message Queues -----
key          msqid        owner        perms        used-bytes   messages
0x160500ff 1             enikop       666          0             0

----- Shared Memory Segments -----
```

SIGHUP hatására az első processz üzenetet helyez az üzenetsorba:

```
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./a.out
PID: 2312
Az uzenetsor létrejött
Uzenet kuldes: SIGHUP, uzenetsor torlese: SIGTERM, varakozas szignalra...
Uzenet elkuldve: A macska aranyos.
Uzenet kuldes: SIGHUP, uzenetsor torlese: SIGTERM, varakozas szignalra...
```

```
█
```

```
enikop@enikop-VirtualBox: ~
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2312
enikop@enikop-VirtualBox:~$ █
```

A második processz SIGHUP hatására kiolvassa az üzenetet:

```
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./b.out
PID: 2311
Uzenet fogadasa: SIGHUP, varakozas szignalra...
Fogadva 19 byte: A macska aranyos.
Uzenet fogadasa: SIGHUP, varakozas szignalra...
```

```
█
```

```
enikop@enikop-VirtualBox: ~
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2312
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ █
```

Ha nincs új kiolvasható üzenet a szignál érkezésekor, a második processz kiírja ezt:

```
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./b.out
PID: 2311
Uzenet fogadasa: SIGHUP, varakozas szignalra...
Fogadva 19 byte: A macska aranyos.
Uzenet fogadasa: SIGHUP, varakozas szignalra...
Nincs uj uzenet, terjen vissza kesobb!
Uzenet fogadasa: SIGHUP, varakozas szignalra...
█
```

```
enikop@enikop-VirtualBox: ~
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2312
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$
```

SIGTERM hatására az első processz törli az üzenetsort és kilép:

```
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./a.out
PID: 2312
Az uzenetsor létrejott
Uzenet kuldesse: SIGHUP, uzenetsor torlese: SIGTERM, varakozas szignalra...
Uzenet elkuldve: A macska aranyos.
Uzenet kuldesse: SIGHUP, uzenetsor torlese: SIGTERM, varakozas szignalra...
Az uzenetsor megszunt, kilepes...
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ █
```

```
enikop@enikop-VirtualBox: ~
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2312
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ kill 2312
enikop@enikop-VirtualBox:~$ █
```

A második processz nem létező üzenetsorból nem tud olvasni, ezt jelzi is, ha véletlenül új SIGHUP-ot kapna a megszűnés után:

```
enikop@enikop-VirtualBox:~/Desktop/OS/Beadando$ ./b.out
PID: 2311
Uzenet fogadása: SIGHUP, varakozas szignalra...
Fogadva 19 byte: A macska aranyos.
Uzenet fogadása: SIGHUP, varakozas szignalra...
Nincs uj uzenet, terjen vissza kesobb!
Uzenet fogadása: SIGHUP, varakozas szignalra...
Az uzenetsor nem letezik.
Uzenet fogadása: SIGHUP, varakozas szignalra...
█

enikop@enikop-VirtualBox: ~
File Edit View Search Terminal Help
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2312
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ kill 2312
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ █
```

```
enikop@enikop-VirtualBox:~$ kill -SIGHUP 2311
enikop@enikop-VirtualBox:~$ ipcs

----- Message Queues -----
key          msqid        owner          perms          used-bytes   messages
-----

----- Shared Memory Segments -----
key          shmid        owner          perms          bytes         nattch       status
0x00000000  6            enikop        600            524288        2            dest
```