

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

**Készítette:** Martinák Mátyás

**Neptunkód:** KLNSPG

# I. Algoritmus

## A feladat leírása:

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: best fit, next 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 a Microsoft Excel programban.

**A legjobban megfelelő (Best fit):** A legkisebbet foglaljuk le azon szabad területek közül, amelyek legalább akkorák, mint a lefoglalandó terület.

**A következő megfelelő (Next fit):** A keresést nem a tárr elejéről kezdjük, hanem az után a terület után, amit legutoljára lefoglaltunk.

Módszerek az eredmények javítására:

- **Szemétygyűjtés (garbage collection)** alkalmazása, vagyis a memória allokáció futás idejű átrendezése nem kielégíthető igények esetén. Hátránya, hogy erőforrás igényes
- **Lapszervezés használata.** Hátránya, hogy kell hozzá MMU (Memory Management Unit) támogatás.
- **Más fajta stratégiát alkalmazni:** pl. A Next fit esetében hatékonyabban ki van használva a memóriefoglalás

## A futtatás eredménye:

Best fit:

BEST FIT					
Igény	Szabad terület				
	50K	30K	200K	16K	30K
20K	50	20, 10	200	16	30
30K	50	30	200	16	30, 0
10K	50	20, 10, 0	200	16	30
100K	50	30	100, 100	16	30
60K	50	30	100, 60, 40	16	30

Next fit:

NEXT FIT					
Igény	Szabad terület				
	50K	30K	200K	16K	30K
20K	20, 30	30	200	16	30
30K	50	30, 0	200	16	30
10K	50	30	10, 190	16	30
100K	50	30	10, 100, 90	16	30
60K	50	30	10, 100, 60, 30	16	30

## II. IPC

### A feladat leírása:

Készítsen olyan C programot, amely egy nevesített csővezetéken keresztül fogad szavakat és ezeket egy fileba menti egy sorszámmal együtt. Illetve egy olyan kliensprogramot, ami ebbe a csőbe adatok helyez el, amit a felhasználotól kér be. A fő program a SIGTERM szignál hatására szüntesse meg a nevesített csövet és lépjen ki.

### A feladat elkészítésének lépései:

A feladat leírás alapján tudjuk, hogy két programra lesz szükségünk, így először létrehoztam a két forrásállományt. (client.c, main.c). Külön függvény hoztam létre a szignál kezelésre, amely a SIGTERM hívás érzékelése után bezárja a file-t, és megpróbálja kitörölni a pipeline-t. Ezek után \_exit()-el kilép a programból.

```
void InterruptHandler(int sig)
{
    fclose(file_out);
    if (remove(KLNSPG) == 0)
    {
        printf("\nPipeline deleted!\n");
    }
    else
    {
        printf("\nCannot deleted pipeline!\n");
    }
    exit(0);
}
```

Ezek után elkezdtem írni a main.c (tehát a főprogram) main függvényét. Létrehoztam benne az mkfifo() hívással a nevesített csővezeték, majd utána megnyitottam az „output.txt” nevű file-t. Megvizsgáltam, hogyha a file nem létezik, akkor -1 -el kidob a program. Ezek után jött a szignálkezelés. Ha a rendszer nem tudja lekezeln a SIGINT-et, akkor a program visszatér 0-val, ellenkező esetben pedig elkezdni kiolvasni a szöveget a pipe-ba.

```
int main()
{
    int fd;
    int count = 0;
    char readbuffer[80];
    mkfifo(KLNSPG, 0666);
    fd = open(KLNSPG, 0666);
    file_out = fopen("output.txt", "w");

    if (file_out == NULL)
    {
        printf("Failed! File does not exist!");
        exit(-1);
    }

    if (signal(SIGINT, InterruptHandler) == SIG_ERR)
    {
        printf("Couldnt set the handler to the \"SIGINT\" signal!\n");
        return 0;
    }
    printf("Starting read the texts writing into the pipe:\n");
```

Ezek után egy végtelen ciklusban valósítottam meg csővezetékbe történő olvasást. Ezzel párhuzamosan, a program elején lévő 'count' integer változó értékét is mindig egyel megnöveltem, hogy a file-ba írásnál sorszámozot kapjon a kiolvasott szöveg. A program a SIGINT, tehát a CTRL+C billentyűkombináció hatására fejezi be a futást.

```
while (1)
{
    read(fd, readbuffer, sizeof(readbuffer));
    count = count + 1;
    fprintf(file_out, "%d. %s\n", count, readbuffer);

    printf("Texts written into the pipe and the file: %s\n", readbuffer);
    sleep(1);
}
return 0;
}
```

A client.c ennél is könnyebben van kitalálva. Mindenek előtt szükségünk van egy 'exitter' nevű boolean változóra, amit az #include <stdbool.h> makróval hoztam be. Amíg ez a változó hamis, addig kéri az inputot a konzolról, illetve ha megkapja az 'exit' parancsot, leáll, ellenkező esetben pedig olvas.

```
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdbool.h>

int main()
{
    int fd;
    bool exitter = false;
    char input[128];

    char *me = "/tmp/KLNSPG";

    fd = open(me, O_WRONLY);
    wait(NULL);
    printf("Write here texts whiches wanted to be seen in the pipeline.\nAfter entering 'exit' you will leave this mode.\n");
    while (!exitter)
    {
        scanf("%s", input);
        if (strcmp(input, "exit") == 0)
        {
            exitter = true;
            close(fd);
            exit(0);
        }
        else
        {
            printf("Entered text: %s\n", input);
            write(fd, input, (strlen(input) + 1));
        }
    }
    return 0;
}
```

A futtatás eredménye:

main.c:

```
matyilinuxsza@matyilinuxsza-VirtualBox: ~/Asztal/OSSemTask_KLNSPG
Fájl Szerkesztés Nézet Keresés Terminál Súgó
matyilinuxsza@matyilinuxsza-VirtualBox:~$ cd Asztal
matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal$ cd OSSemTask_KLNSPG/
matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal/OSSemTask_KLNSPG$ ./main
Starting read the texts writing into the pipe:
Szia
Uram
Ez
EGy
Operacios
rendszerek
TASK
^C
Pipeline deleted!matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal/OSSemTask_KLNSP
G$
```

client.c:

```
matyilinuxsza@matyilinuxsza-VirtualBox: ~/Asztal/OSSemTask_KLNSPG
Fájl Szerkesztés Nézet Keresés Terminál Súgó
matyilinuxsza@matyilinuxsza-VirtualBox:~$ cd Asztal
matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal$ cd OSSemTask_KLNSPG/
matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal/OSSemTask_KLNSPG$ ./client
Write here texts whiches wanted to be seen in the pipeline.
After entering 'exit' you will leave this mode.
Szia
Entered text: Szia
Uram
Entered text: Uram
Ez
Entered text: Ez
EGy
Entered text: EGy
Operacios rendszerek
Entered text: Operacios
Entered text: rendszerek
TASK
Entered text: TASK
exit
matyilinuxsza@matyilinuxsza-VirtualBox:~/Asztal/OSSemTask_KLNSPG$
```