

# **Operációs rendszerek BSc**

## **10. Gyak.**

2022. 04. 13.

### **Készítette:**

Gyáni Kevin Zsolt Bsc

Szak Programtervező informatikus

Neptunkód CBOYZF

**Miskolc, 2022**

### **1.feladat**

A bankgáralgortimus megoldás, levezetéssel az excel táblázatban található!

## 2.feladat

```
#include <string.h>

int main() {
    int pipefd[2];    // 2 eleme tomb, mert ket vege lesz a csonek, ezert ket változó neki
    pid_t cpid;      // gyermek pidje lesz benne
    char buf;        // bufferke
    char szoveg[32];  // másik buffer a küldendő szövegnek
    int ret;

    if (pipe(pipefd) == -1) { // krealunk csövezeteket, és a két fd a pipefd-ben lesz tárolva
        perror("pipe");
        exit(-1);
    }

    printf("%d: fd1: %d, fd2: %d\n", getpid(), pipefd[0], pipefd[1]);

    cpid = fork();
    if (cpid == -1) {
        perror("fork");
        exit(-1);
    }

    if (cpid == 0) {
        /* gyerek vagyok */
        printf("%d: szülő vagyok\n", getpid());

        close(pipefd[0]); // itt meg az olvasó végét csukjuk le, mert írni fogunk */

        strcpy(szoveg, "ez itt kerem egy szöveg\0"); // feltöltöm a buffert a küldeni kívánt stringgel
        printf("%d: küldöm a szöveget: %s:%d\n", getpid(), szoveg, strlen(szoveg));

        write(pipefd[1], szoveg, strlen(szoveg)); // ontunk anyagot a csőbe

        close(pipefd[1]); // lecsukom a csatorna fedelet, a másik oldalon EOF lesz */

        wait(NULL); // várjunk gyerekre */

        exit(0);
    } else { // szülő vagyok */

        printf("%d: gyerek vagyok\n", getpid());
        close(pipefd[1]); // a gyermek csak olvasni fog, ezért az írás fd-t le is csukom */

        printf("%d: megnézem mi van a csőben\n%d: ", getpid(), getpid());
        while (read(pipefd[0], &buf, 1) > 0) { // olvasunk a cső kimenetéről, amíg van mit (1 byteonként)
            printf("%c", buf);
        }
        printf("\n%d: lezártak a cső másik oldalát\n", getpid());

        close(pipefd[0]); // na itt becsukom már az olvasó végét is */

        exit(0); // kilep
    }
}
```

Gyerek: bezárjuk az olvasó végét a csőnek `close(pipefd[0]);` majd a `write(pipefd[1], szoveg, strlen(szoveg))` függvénnyel írunk a csőbe. Majd az írás végét is bezárjuk. Felnőtt:

bezárjuk az íróvégét `close(pipefd[1]);` majd a `while read-el` betűnként beolvassuk a csőben lévő szöveget. Utána bezárjuk a cső olvasóvégét is majd `exit()`-el kilépünk.

### 3.feladat

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/file.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>

int main()
{
    int fd, ret;
    char buf[32];

    buf[0]=0;

    ret=mkfifo("fifoka",00666);    // létrehozom a fifo-t a filerendszeren
    if (ret == -1) {
        perror("mkfifo()");
        exit(-1);
    }

    fd=open("fifoka",O_RDWR);      // megnyitom a fifot amit az iment hoztam létre
    if (fd == -1) {
        perror("open() hiba!");
        exit(-1);
    }

    strcpy(buf,"ezt teszem a fifoba\0");
    printf("írok a fifoba: %s:%d\n",buf,strlen(buf));
    write(fd,buf,strlen(buf)); // írok bele valamit, hogy ne legyen üres

    ret=read(fd,buf,32); // olvasok belőle ugyanazt, ret: mennyit sikerült olvasni
    printf("read() olvasott %d byteot, ami a következő: %s\n",ret,buf);

    close(fd);

    unlink("fifoka");    // letörölöm a fifomat az fs-ról
}
```

Létrehozás: mkfifo(„név”,id) fd=open(„név”,O\_RDWR); kinyitom a csövet írásra és olvasásra. write() írok bele. read() olvasok belőle. close()-al bezárom, unlink()-el pedig törölöm