

JEGYZŐKÖNYV

Operációs rendszerek BSc

2021 tavasz féléves feladat

Készítette: **Nagy Róbert**

Neptunkód: **JMDRGG**

A feladat leírása:

6.feladat:

Írjon C nyelvű programokat, amelyek létrehoznak egy nevesített csővezetékét. Az egyik program megnyitja, olvasni próbál belőle. Közben egy másik program ír bele egy véletlen számot. Az első program kiírja a kapott számot és visszaküldi ennek a számnak a kétszeresét. A másik program kiolvassa és kiírja a szám új értékét. A folyamatok megszűnnek és a második program eltünteti a nevesített csővezetékét.

A feladat elkészítésének lépései:

1.Lépés: Létrehozunk egy Szerver és Kliens programot. A szerver program olvasni fogja a kliens program által küldött véletlen számot.

```
void main() {
    int pipe;
    //megnezi, hogy nincs fifo, ennek normalis mukodeskor
    if (access(CSONEV, F_OK)) {
        pipe = mkfifo(CSONEV, S_IFIFO | 0666);
    }

    int fd = open(CSONEV, O_RDWR);
    if (fd == -1) {
        perror("Szerver : nem sikerult megnyitni \n");
        exit(-1);
    }

    olvasMajdVisszair(fd);
    close(fd);
    printf("\nbefejezve\n");

    unlink(CSONEV);
}
```

2.Lépés: A szerver program létrehozza a nevesített csővezetékét. A szerver az access() hívása segítségével megnezi, hogy létezik-e adott nevű csővezeték, ha nem létezik, akkor létrehozza.

3.Lépés: A csővezeték megnyitása.

4.Lépés: olvasMajdVisszair() metódus létrehoz egy változót, amelybe a kapott értéket tároljuk. Itt a szerver program meghívja a read system callt. Itt a szerver program várni fog amíg a csővezetékbe nem kerül érték.

```

void olvasMajdVisszair(int fd) {
    int val = 0;

    printf("szerver olvas..\n");
    if (read(fd, &val, sizeof(val)) < 0) {
        perror("Szerver : nem sikerult olvasni\n");
        exit(-1);
    }

    printf("szerver megkapta : %d \n", val);
    val *= 2;

    if (write(fd, &val, sizeof(val)) < 0) {
        perror("Szerver(2) : nem sikerult irni\n");
        exit(-1);
    }
}

```

5.Lépés: Kliens program elkészítése. A kliens program megnyitja a csővezetékét és egy véletlen számot fog generálni és azt beleírja.

```

void main() {
    srand(time(0));

    int fd = open(CSONEV, O_RDWR);
    if (fd == -1) {
        perror("kliens : nem sikerult megnyitni \n");
        exit(-1);
    }

    printf("kliens indul...\n");
    kliensIr(fd);
    kliensOlvas(fd);
    close(fd);
    unlink(CSONEV);
}

```

6.Lépés: Kliens ír egy véletlen számot a write rendszerhívás segítségével. Az írás után a szerver program folytatódik.

```

void kliensIr(int fd) {
    int n = getRandom();
    if (write(fd, &n, sizeof(int)) == -1) {
        perror("kliens : Nem sikerult irni\n");
        exit(-1);
    }
    printf("kliens : irva\n");
}

```

```

int getRandom() {
    return (rand() % 10) + 1;
}

```

7.Lépés: A szerver program megszorozza a kapott értéket 2-vel, majd visszaírja a csővezetékbe.

```

void olvasMajdVisszair(int fd) {
    int val = 0;

    printf("szerver olvas..\n");
    if (read(fd, &val, sizeof(val)) < 0) {
        perror("Szerver : nem sikerult olvasni\n");
        exit(-1);
    }

    printf("szerver megkapta : %d \n", val);
    val *= 2;

    if (write(fd, &val, sizeof(val)) < 0) {
        perror("Szerver(2) : nem sikerult irni\n");
        exit(-1);
    }
}

```

8.Lépés: A kliens program kiolvassa a megduplázott értéket, majd kiírja standard outputra.

```

2
3 void kliensOlvas(int fd) {
4     int val;
5     printf("kliens olvas...\n");
6     if (read(fd, &val, sizeof(val)) < 0) {
7         perror("kliens : Nem sikerult olvasni\n");
8         exit(-1);
9     }
10    printf("\nkliens kapott ertek : %d\n", val);
11 }

```

9. Lépés: A kliens és a szerver is unlinkeli a csővezetékét, ezzel kitörölve azt.

A futtatás eredménye:

```
→ testing ./jmdrgg_feleves_szerver
szerver olvas..
szerver megkapta : 10

befejezve
→ testing
```

```
→ testing ./jmdrgg_feleves_kliens
kliens indul...
kliens : irva
kliens olvas...

kliens kapott ertek : 20
→ testing |
```