

Operációs rendszerek BSc

10. Gyak.

2022. 04. 12.

Készítette:

Sziráczki Soma

Bsc

Programtervező

informatikus

BK6QE8

Miskolc, 2022

1. Az előadáson bemutatott mintaprogram alapján készítse el a következő feladatot. Adott egy rendszerbe az alábbi erőforrások: R (R1: 10; R2: 5; R3: 7) A rendszerbe 5 processz van: P0, P1, P2, P3, P4 Kérdés: Kielégíthető-e P1 (1,0,2), P4 (3,3,0) ill. P0 (0,2,0) kérése úgy, hogy biztonságos legyen, holtpontmentesség szempontjából a rendszer - a következő kiinduló állapot alapján. Külön-külön táblázatba oldja meg a feladatot!

- Határozza meg a processzek által igényelt erőforrások mátrixát?
- Határozza meg pillanatnyilag szabad erőforrások számát?
- Igazolja, magyarázza az egyes processzek végrehajtásának lehetséges sorrendjét - számolással?

Megvalósítás:

Az összes osztály-erőforrások száma: (10,5,7)											
Kiinduló állapot											
1. Lépés				2. Lépés				3. Lépés			
MAX IGÉNY				FOGLAL				IGÉNY = MAX IGÉNY - FOGLAL			
	R1	R2	R3		R1	R2	R3		R1	R2	R3
P0	7	5	3	P0	0	1	0	P0	7	4	3
P1	3	2	2	P1	2	0	0	P1	1	2	2
P2	9	0	2	P2	3	0	2	P2	6	0	0
P3	2	2	2	P3	2	1	1	P3	0	1	1
P4	4	3	3	P4	0	0	2	P4	4	3	1

MAXr = (10, 5, 7)
Szabad = (10, 5, 7) - (7, 2, 5) = (3, 3, 2)
P4 Szabad = [3, 3, 2] + [3, 3, 0] = [6, 6, 2]
P0 Szabad = [6, 6, 2] + [0, 2, 0] = [6, 8, 2]

2. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezeték, 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

Megvalósítás:

```

sziraczki@sziraczki-VirtualBox:~$ cd Asztal
sziraczki@sziraczki-VirtualBox:~/Asztal$ cd forras3
sziraczki@sziraczki-VirtualBox:~/Asztal/forras3$ gcc -o BK6QE8_unnamed.o BK6QE8_unnamed.c
sziraczki@sziraczki-VirtualBox:~/Asztal/forras3$ ./BK6QE8_unnamed.o
Sziraczki Soma BK6QE8sziraczki@sziraczki-VirtualBox:~/Asztal/forras3$

```

```

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

int main() {
    int fd[2];
    int child;

    if(pipe(fd)) {
        perror("pipe");
        return 1;
    }

    child = fork();

    if(child > 0) {
        char s[1024];
        close(fd[1]);
        read(fd[0], s, sizeof(s));
        printf("%s", s);

        close(fd[0]);
    } else if (child == 0) {
        close(fd[0]);
        write(fd[1], "Sziraczki Soma BK6QE8\n", 21);
        close(fd[1]);
    }

    return 0;
}

```

3. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékét (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl.: Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: neptunkod_named.c

Megvalósítás:

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

int main() {
    int child;

    mkfifo("Sziraczki Soma", S_IRUSR | S_IWUSR);
    child = fork();

    if(child > 0) {
        char s[888];
        int fd;

        fd = open("Sziraczki Soma", O_RDONLY);
        read(fd, s, sizeof(s));
        printf("%s", s);
        close(fd);
        unlink("Sziraczki Soma");
    } else if(child == 0) {
        int fd = open("Sziraczki Soma", O_RDONLY);
        write(fd, "SZS BK6QE8\n", 12);
        close(fd);
    }
    return 0;
}
```