

Operációs rendszerek BSc

10.Gyak.

2022. 04.12.

Készítette:

Ónodi Bence BSC

Programtervező
informatikus

RYSNLC

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!

a) Határozza meg a processzek által igényelt erőforrások mátrixát?

b) Határozza meg pillanatnyilag szabad erőforrások számát?

c) Igazolja, magyarázza az egyes processzek végrehajtásának lehetséges sorrendjét - számolással?"

Az összes erőforrások száma: (10,5,7)											
Kiinduló állapot				P4 {3,3,0} kérése							
1. lépés				2. lépés							
MAX IGÉNY				FOGLAL							
	R1	R2	R3		R1	R2	R3	Kielégítetlen igények		max-foglal	
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	3	3	2	p4	1	0	1

2. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékét, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

```
#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], "Onodi Bence RYSNLC\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.

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

int main() {
    int child;

    mkfifo("Onodi Bence", S_IRUSR | S_IWUSR);
    child = fork();

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

        fd = open("Onodi Bence", O_RDONLY);
        read(fd, s, sizeof(s));
        printf("%s", s);
        close(fd);
        unlink("Onodi Bence");
    } else if(child == 0) {
        int fd = open("Onodi Bence", O_RDONLY);
        write(fd, "OB RYSNLC\n", 12);
        close(fd);
    }
    return 0;
}
```