

Operációs rendszerek

Hetedik gyakorlat

Készítette: Énekes Zoltán László

1. Adott négy processz a rendszerbe, melynek beérkezési sorrendje: A, B, C és D. Minden processz USER módban fut és mindegyik processz futásra kész.

- Határozza meg az ütemezést RR nélkül és az ütemezést RR-nal - külön-külön táblázatba.
- Minden óráutem esetén határozza meg a processzek sorrendjét óráutés előtt/után.
- Igazolja a számítással a tanultak alapján.

Round Robin		A process		B process		C process		D process		Reschedule	
Clock tick		p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	running_before	running_after
Starting point		60	0	60	0	60	0	60	0	A	A
1		60	1	60	0	60	0	60	0	A	A
2		60	2	60	0	60	0	60	0	A	A
9		60	9	60	0	60	0	60	0	A	A
10		60	10	60	0	60	0	60	0	A	B
11		60	10	60	1	60	0	60	0	B	B
20		60	10	60	10	60	0	60	0	B	C
21		60	10	60	10	60	1	60	0	C	C
30		60	10	60	10	60	10	60	0	C	D
31		60	10	60	10	60	10	60	1	D	D
39		60	10	60	10	60	10	60	9	D	A
40		60	10	60	10	60	10	60	10	D	A
50		60	20	60	10	60	10	60	10	A	B
60		60	20	60	20	60	10	60	10	B	C
70		60	20	60	20	60	20	60	10	C	D
80		60	20	60	20	60	20	60	20	D	A
90		60	30	60	20	60	20	60	20	A	B
100		60	24	66	24	64	16	78	16	C	C
150		60	24	66	24	64	66	78	16	C	C
190		60	24	66	24	64	106	78	16	C	C
199		60	24	66	24	64	115	78	16	C	C
200		60	16	64	16	79	77	90	11	C	C
201		60	17	64	16	79	77	90	11	A	C

Nem Round Robin		A process		B process		C process		D process		Reschedule	
Clock tick		p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	running_before	running_after
Starting point		60	0	60	0	60	0	60	0	A	A
1		60	0	60	0	60	0	60	0	A	A
2		60	0	60	0	60	0	60	0	A	A
3		60	0	60	0	60	0	60	0	A	A
10		60	10	60	0	60	0	60	0	A	A
11		60	11	60	0	60	0	60	0	A	B
99		60	99	60	0	60	0	60	0	A	B
100		73	50	60	0	60	0	60	0	A	B
101		73	50	60	1	60	0	60	0	B	B
199		73	50	60	99	60	0	60	0	B	B
200		66	25	73	50	60	0	60	0	B	C
201		66	25	73	50	60	1	60	0	C	C

2. A tanult rendszerhívásokkal (`open()`, `read()/write()`, `close()` - ők fogják a rendszerhívásokat tovább hívni.) írjanak egy `neptunkod_openclose.c` programot, amely megnyit egy fájlt – `neptunkod.txt`, tartalma: hallgató neve, szak , neptunkod.

A program következő műveleteket végezze:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>

int main(void) {
    int fd;
    char *buf[100];
    int len;

    if ((fd = open("CIJA2K.txt", O_RDWR)) < 0) {
        perror("open hiba");
        return -1;
    }

    if ((len = read(fd, buf, sizeof(buf))) != 0) {
        if (len == -1 && errno != EINTR) {
            perror("read hiba");
            return -1;
        }
        int index;
        for (index = 0; index < sizeof(buf) && buf[index] != EOF; index++);
        buf[index] = '\0';
        printf("Beolvasva: %d byte\n Tartalma: %s\n", len, buf);
    }

    if (lseek(fd, 0, SEEK_SET) < 0) {
        perror("lseek hiba");
        return -1;
    }
    printf("Pozicio a fajl elejere mozgatva\n");

    if ((len = write(fd, "Hello World", 11)) != 11) {
        perror("write hiba");
        return -1;
    }
    printf("%d byte kiírva\n", len);

    close(fd);
    return 0;
}
```