

OP.Rendszerek 7.gyak

RR nélkül	A process		B process		C process		D process		Újraütemezés	
Óraütés	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	Ütemezés előtt	Ütemezés után
Kezdő pont	60	0	60	0	60	0	60	0		
1	60	1	60	0	60	0	60	0	A	A
2	60	2	60	0	60	0	60	0	A	A
3	60	3	60	0	60	0	60	0	A	A
...
100	75	50		1	60	0	60	0		
101	75	50		2	60	0	60	0		
102	75	50		3	60	0	60	0		
...
200	75	25	75	50	60	1	60	0		
201	75	25	75	50	60	2	60	0		

Minden process prioritása p_uspri=60 RR nélkül a processzek A,B,C,D sorrendben haladnak. Minden 100-adik clock ticknél ütemezés történik.

RR	A process		B process		C process		D process		Újraütemezés	
Óraütés	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	Ütemezés előtt	Ütemezés után
Kezdő pont	60	0	60	0	60	0	60	0		
1	60	1	60	0	60	0	60	0	A	A
2	60	2	60	0	60	0	60	0	A	A
3	60	3	60	0	60	0	60	0	A	A
...
10	60	10	60	1	60	0	60	0	A	B
11	60	10	60	2	60	0	60	0	B	B
12	60	10	60	3	60	0	60	0	B	B
...
20	60	10	60	10	60	1	60	0	B	C
21	60	10	60	10	60	2	60	0	C	C
22	60	10	60	10	60	3	60	0	C	C
...
30	60	1	60	10	60	10	60	1	C	D
31	60	10	60	10	60	10	60	2	D	D
32	60	10	60	10	60	10	60	3	D	D
...
40	60	11	60	10	60	10	60	10	D	A
41	60	12	60	10	60	10	60	10	A	A
42	60	13	60	10	60	10	60	10	A	A
...
50	60	20	60	11	60	10	60	10	A	B
51	60	20	60	12	60	10	60	10	B	B
52	60	20	60	13	60	10	60	10	B	B
...
60	60	20	60	20	60	11	60	10	B	C
61	60	20	60	20	60	12	60	10	C	C
...
70	60	20	60	20	60	20	60	11	C	D
71	60	20	60	20	60	20	60	12	D	D
...
80	60	21	60	20	60	20	60	20	D	A
81	60	22	60	20	60	20	60	20	A	A
...
90	60	30	60	21	60	20	60	20	A	B
91	60	30	60	22	60	20	60	20	B	B

90	60	30	60	21	60	20	60	20	A	B
91	60	30	60	22	60	20	60	20	B	B
...
100	60	30	60	30	60	21	60	20	B	C
101	60	30	60	30	60	22	60	20	C	C
...
110	60	30	60	30	60	30	60	21	C	D
111	60	30	60	30	60	30	60	22	D	D
...
120	60	31	60	30	60	30	60	30	D	A
121	60	32	60	3	60	30	60	30	A	A
...
130	60	40	60	31	60	30	60	30	A	B
131	60	40	60	32	60	30	60	30	B	B
...
140	60	40	60	40	60	31	60	30	B	C
141	60	40	60	40	60	32	60	30	C	C
...
150	60	40	60	40	60	40	60	31	C	D
151	60	40	60	40	60	40	60	32	D	D
...
160	60	41	60	40	60	40	60	40	D	A
161	60	42	60	40	60	40	60	40	A	A
...
170	60	50	60	41	60	40	60	40	A	B
171	60	50	60	42	60	40	60	40	B	B
...
180	60	50	60	50	60	41	60	40	B	C
181	60	50	60	50	60	42	60	40	C	C
...
190	60	50	60	50	60	50	60	41	C	D
191	60	50	60	50	60	50	60	42	D	D
...
200	60	51	60	50	60	50	60	50	D	A
201	60	52	60	50	60	50	60	50	A	A

Minden 100-adik clock tickre van ütemezés és korrekciós faktor. A 100.lépésnél a kor. faktro:

-kf= $2*2/(2*2+1)$. A 200. lépésnél pedig $2*1/(2*1+1)$. A D processz esetén a p_nice=5, azaz egy kicsit alacsonyabb a prioritása mint a többi processzé.

```

int main(int argc, char const *argv[]) {
    int fileDescriptor = open("jmdrgg.txt", O_RDWR);

    if (fileDescriptor == -1) {
        fprintf(stderr, "Hibas file name");
        return 1;
    }

    char text[256];
    int numberOfBytes = read(fileDescriptor, text, sizeof(text));

    if (numberOfBytes == 0 || numberOfBytes == -1) {
        fprintf(stderr, "Hiba volt a beolvasassal");
    }

    text[numberOfBytes] = '\0';

    printf("Beolvasott szoveg: %s\nByteok szama: %d", text, numberOfBytes);

    lseek(fileDescriptor, 0, SEEK_SET);

    char szoveg[] = "szoveg";
    numberOfBytes = write(fileDescriptor, szoveg, sizeof(szoveg));

    if (numberOfBytes == 0 || numberOfBytes == -1) {
        fprintf(stderr, "Hiba volt a fileba irassal");
    }

    printf("\nkiirt szoveg: %s\nByteok szama: %d", szoveg, numberOfBytes);

    close(fileDescriptor);

    return 0;
}

```

A program system callokkal beolvas és kiír a megadott fileból.