Operációs rendszerek BSc

7.gyak.

2021. 03. 23.

Készítette: Nagy Róbert

Programtervező Informatikus Neptunkód:JMDRGG

Miskolc, 2021

		20									
clock tick	A				C		D		reschedule		
	p_usrpri	р_сро	p_usrpri	p_cpu	p_usrpri	p_cpu	p_userpri	p_cpu	running_before	running_after	
tarting point	60	0	60	-0	60	0	60	0	A	A	P_USER = 60
1	60	1	60	0	60	0	60	0	А	A	p_pri = P_USER + p_cpu / 4 + 2 * p_nice
2	60	2	60	0	60	0	60	0	A	A	
3	60	3	60	0	60	0	60	0	A	A	
	-	-	300	940	7	-10			***		
99	60	99	60	0	60.	0	60	0	A	A	
100	73	50	60	0	60	0	60	0	A	8	
101	73	50	60	11	60	0	60	0		В	
102	23	50	60	2	60	0	60	0	B	6	
0.75	- 90		775.00	10000	- 75	777	100	-	777	-	
199	73	50	60	99	60	0	60	0	#	8	
200	66	25	73	50	60	0	60	0	8	C	
201	66	25	73	50	60	1	60	0	C	C	

Minden processz prioritása p_usrpri = 60. Round robin nélkül a processzek A, B, C, D sorrendben haladnak. Minden 100-adik clock ticknél ütemezés történik.

	Α.				Round Robinos		n.		coschedulo				
look High		A coul	in arrange	B	a special Control	C CON	D D	ener:		schedule	iar.		
ock tick	p_usepei	p_cpu	p_usrpri	p_cpu	p_usepri	p_cpu	p_userpri p	cpu .		ore running_af	er	0.0000.00	
arting point	60	0	60	0	60	0	60	0	A	A		P_USER = 60	100 14 17 to ole
1 2		1 2	60	0	60 60	0	60	0	A	A		p_pn = P_USER + p	_cpu / 4 + 2 * p_nice
	.00	4.5				U		· ·				Diesetén a pi mice -	. 61
9	60	9	60	0	60	0	60	0	A			Diezereu a b_urce .	- 34
10		10	60	D	60	0	60	0	A	8			
11		10	60	1	60	0	60	0	B	8			
12		10	60	2	60	0	60	0	8	8			
				55.0									
19		10	60	9	60	0	60	0		8			
20		10	60	10	60	0	60	0		c			
21		10	60	10	50	1	60	0	c	c			
22		10	60	10	60	2	60	0	c	è			
						- 8		1					
29		10	60	10	50	9	50	0	c	c			
30		10	60	10	50	10	.60	0	c	D			
31		10	60	10	60	10	60	11	D	D			
32		10	60	10	50	10	60	2	Đ	D			
		200	211	100		-							
39	60	10	60	10	60	10	60	9	D	D			
40		10	60	10	60	10	60	10	D	А			
			340	1441	1.000		1000			140			
50	60	20	60	10	60	10	60	10	A	8			
	200	***	101		Test-		- 100		5.000	-1946	- 040	100	50.000
	60	60	20		60	20	60		10	60	10	В	C
			22			-	92.0		200	444	***		
	70	60	20		60	20	60		20	60	10	C	D
	10	ou				2.0	.00		2.U	.00	10		
	-340	***	+++		144	244	***		***	(1)	-		***
	80	60	20		60	20	60		20	60	20	D	A
	- 100		77		100	-	100		***	191	000	940	***
	90	60	30		60	20	60		20	60	20	A	В
		222	***				***			***	***		***
	99	60	30		60		60						
						29			20	60	20	В	В
	100	66	24		66	24	64		16	78	16	В	C
		22	***			-	1700		1	1447			
	110	66	24		66	24	64		26	78	16	C	C
			***		444	1000	77.996		1199	***	***	***	***
	120	66	24		66	24	64		36	78	16	c	c
	120	ou			00	2.4	100		30		10		
	-110	***	100		140	200	1.000		1.000	1,555	440	***	1991
	130	66	24		66	24	64		46	78	16	C	C
	***	440	***		100	200	200		***	494	200	***	100
	140	66	24		66	24	64		56	78	16	C	C
	144	144	147			Take	199449		1744	test.	444	***	***
	150	66	24		66	24	64		66	78	16	C	c
	-	jest.	+++		1994		(1.444)				***	D 100	
	160	66	24		66	24	64		76	78	16	c	С
	100	00	24		00	2.4	04			10	10		
		***	797			900	440		1.000	TTT:	TTY.	1999	***

```
C
                                                                                               c
                                 24
199
                24
                                 24
                                                                     16
                                                                                                                KF = 2/3
200
                                 16
                                                                     11
                16
                                                   77
201
                                 16
                                          79
                                                   77
                                                            90
```

Minden 100-adik clock tickre van ütemezés és korrekciós faktor. A 100. lépésnél a korrekciós faktor: 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", 0_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 callok segítségével beolvas és kiír a megadott filebol.