

Operációs rendszerek BSc

3. Gyak.

2022. 05. 12.

Készítette:

Juhász Tibor

Mérnökinformatikus BSc. levelező

X7KWVG

1, Feladat leírás:

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

Kezdetben mindegyik processz $p_uspri = 60$.

Az A, B, C processz $p_nice = 0$, a D processz $p_nice = 5$.

Mindegyik processz $p_cpu = 0$, az óráütés 1 indul, a befejezés legyen 201. óráütés-ig.

a.) Határozza meg az ütemezést RR nélkül és az ütemezést RR-nal - külön-külön táblázatba.

b.) Minden óráütem esetén határozza meg a processzek sorrendjét óráütés előtt/után.

c.) Igazolja a számítással a tanultak alapján.

Megoldás:

a, és b,

RR nélkül:

RR nélkül	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
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
...
99	60	99	60	0	60	0	60	0	A	A
100	$60+50/4$ 72	$100/2$ 50	60	0	60	0	60	0	A	B
101	75	50	60	1	60	0	60	0	B	B
102	75	50	60	2	60	0	60	0	B	B
...
199	75	50	60	99	60	0	60	0	B	B
200	$60+25/4$ 66	$50/2$ 25	$60+50/4$ 72	$100/2$ 50	60	0	60	0	B	C
201	66	25	75	50	60	1	60	0	C	C

RR-al:

RR	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
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
...
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
...
19	60	10	60	9	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
...
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+6 66	30*0,86 26	60+6 66	30*0,86 26	60+4 64	20*0,86 17	60+4+10 74	20*0,86 17	B	C
...
110	66	26	66	26	64	27	74	17	C	C
...
120	66	26	66	26	64	37	74	17	C	C
...
130	66	26	66	26	64	47	74	17	C	C
...
140	66	26	66	26	64	57	74	17	C	C
...
150	66	26	66	26	64	67	74	17	C	C
...
160	66	26	66	26	64	77	74	17	C	C
...
170	66	26	66	26	64	87	74	17	C	C
...
180	66	26	66	26	64	97	74	17	C	C
...
190	66	26	66	26	64	107	74	17	C	C
...
200	60+5 65	26*0,8 21	60+5 65	26*0,8 21	60+22 82	107*0,8 86	60+13 73	17*0,8 14	C	A
201	65	22	65	21	82	86	73	14	A	A

c,

RR nélkül:

Az A processz 100-ig fog futni. Mivel csak ő futott, ebben az esetben csak nála szükséges számolnunk.

$$p_cpu = p_cpu * KF = 100/2 = 50$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 50/4 = 72$$

Innentől 200-ig a B processz fog futni.

200-nál ismét szükséges számolnunk, itt már az A-nál és B-nél is.

A processz:

$$p_cpu = p_cpu * KF = 50/2 = 25$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 25/4 = 66$$

B processz:

$$p_cpu = p_cpu * KF = 100/2 = 50$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 50/4 = 72$$

201-nél a C processz fog futni.

RR-al:

A processzek 10 óraütést futnak, utána a következő azonos/alacsonyabb prioritású processz fog futni.

100-ig nem szükséges számolnunk.

100-nál mind a négy processznél szükséges számolnunk, mivel mind a négy futott már.

$$KF = 2*3/2*3+1 = 0,86$$

A processz:

$$p_cpu = p_cpu * KF = 30 * 0,86 = 26$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 26/4 = 66$$

B processz:

$$p_cpu = p_cpu * KF = 30 * 0,86 = 26$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 26/4 = 66$$

C processz:

$$p_cpu = p_cpu * KF = 20 * 0,86 = 17$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 17/4 = 64$$

D processz:

$$p_cpu = p_cpu * KF = 20 * 0,86 = 17$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2*p_nice = 60 + 17/4 + 2*5 = 74$$

Innentől a 200. óraütésig a legalacsonyabb prioritású processz a C lesz, végig ő fog futni.

$$64 < 66 < 74$$

A 200. óraütésnél kell újra számolnunk.

$$KF = 2 \cdot 3 / 2 \cdot 3 + 1 = 0,86$$

A processz:

$$p_cpu = p_cpu \cdot KF = 26 \cdot 0,86 = 21$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2 \cdot p_nice = 60 + 21/4 = 65$$

B processz:

$$p_cpu = p_cpu \cdot KF = 30 \cdot 0,86 = 21$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2 \cdot p_nice = 60 + 21/4 = 65$$

C processz:

$$p_cpu = p_cpu \cdot KF = 107 \cdot 0,86 = 86$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2 \cdot p_nice = 60 + 86/4 = 82$$

D processz:

$$p_cpu = p_cpu \cdot KF = 17 \cdot 0,86 = 14$$

$$p_uspri = P_USER \text{ (esetünkben konstans 60) } + p_cpu/4 + 2 \cdot p_nice = 60 + 14/4 + 2 \cdot 5 = 73$$

Most a legalacsonyabb prioritású processz az A lesz, emiatt ő fog futni.

3, Feladat leírás:

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.

Mentés: neptunkod_unnamed.c

Megoldás:

A program megírása után terminálban fordítom és futtatom.

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_unnamed X7KWVG_unnamed.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_unnamed
Olvasas eredménye: Juhasz Tibor X7KWVG
tibi@tibi-VirtualBox:~$
```

A programban a szülő processz létrehoz egy csővezetékét, a gyermek processz beleírja a „Juhasz Tibor X7KWVG” sztringet, amit a szülő processz kiolvas és kiírja a standard kimenetre.

4, Feladat leírás:

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

Megoldás:

A program megírása után terminálban fordítom és futtatom.

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_named X7KWVG_named.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_named
Olvasas eredménye: Juhasz Tibor
tibi@tibi-VirtualBox:~$
```

A szülő processz létrehoz egy X7KWVG nevű csővezeték, a gyerek processz beleírja a „Juhasz Tibor” sztringet. A szülő kiolvassa és kiírja a standard kimenetre.

5, Feladat leírás:

Adott egy rendszerbe az összes osztály-erőforrások száma: R (R1: 10; R2: 9; R3: 12)

A rendszerbe 4 processz van: P1, P2, P3, P4.

Biztonságos-e holtpontmentesség szempontjából a rendszer - a következő kiinduló állapot alapján?

- a) Határozza meg a folyamatok által igényelt erőforrások mátrixát?
- b) Határozza meg pillanatnyilag szabad erőforrások számát?
- c) Igazolja az egyes processzek végrehajtásának lehetséges sorrendjét - számolással?”

	MAX.IGÉNY					FOGLAL		
	R1	R2	R3			R1	R2	R3
P1	4	4	5		P1	2	2	3
P2	1	4	3		P2	1	2	2
P3	6	7	7		P3	0	1	3
P4	3	7	10		P4	2	1	2

Feladat megoldása:

IGÉNY meghatározása.

$IGÉNY = MAX.IGÉNY - FOGLAL$

IGÉNY		
R1	R2	R3
2	2	2
0	2	1
6	6	4
1	6	8

Szabad erőforrások száma:

Összes erőforrás – FOGLAL (adott erőforrásból). Pl.: R1: 10 az összes. 5 a FOGLAL, tehát $10 - 5 = 5$

KÉSZLET: (5, 3, 2)

Következő lépésként megnézzük, hogy van-e olyan processz, amelynek az igénye kielégíthető.

A P1 processz igénye kielégíthető, mivel az igénye (2, 2, 2), a készletünk pedig (5, 3, 2).

P1 futása után az általa foglalt erőforrás felszabadul, az új készletünk: (7, 5, 5)

Most a P2 processz is kielégíthető.

A P2 processz lefut.

P2 futása után az általa foglalt erőforrás felszabadul, az új készlet: (8, 7, 7)

A P3 processz igénye is kielégíthető.

Lefut a P3 processz.

P3 futása után az új készlet: (8, 8, 10)

Ezután kielégíthető a P4 processz is.

A P4 processz lefut, és minden erőforrás felszabadul.