

Zadatak

U promatranom trenutku stanje sustava je sljedeće: dretva 7 je aktivna; dretve 5, 3 i 1 su u redu pripravnih dretvi, dretva 4 je u redu općeg semafora S te dretva 6 je u redu odgođenih – treba još 5 otkućaja sata da se od tamo makne. Svi redovi organizirani su po prioritetu gdje veći broj dretve označava veći prioritet (osim odgođenih i redova naprava). Pozivi koji se tada događaju su redom, jedan nakon drugoga: `PostaviOSEM(S)`, `ZapočniUI(1)`, `Odgodi(7)` te `Prekid_sata()`. Pokazati stanje sustava nakon svakog od ovih poziva.

Rješenje (sadržaj tablice)

	Početno stanje	Nakon <code>PostaviOSEM(S)</code>	Nakon <code>ZapočniUI(1)</code>	Nakon <code>Odgodi(7)</code>	Nakon <code>Prekid_sata()</code>
Aktivna_D	7	7	5	4	4
Pripravne_D	5 3 1	5 4 3 1	4 3 1	3 1	3 1
Odgođene_D	6 ⁵	6 ⁵	6 ⁵	6 ⁵ 5 ²	6 ⁴ 5 ²
OSEM[S]	4 (.v=0)	- (.v=0)	- (.v=0)	- (.v=0)	- (.v=0)
Red UI[1]	-	-	7	7	7

Zadatak

Problem sličan problemu pet filozofa pojavljuje se u „Restoranu na kraju svemira“. Tamo za istim stolom sjede osobe s različitim brojem ruku i različitim običajima hranjenja. Pretpostaviti da se na sredini stola nalazi 20 vilica i da svaka osoba koja jede mora u svakoj ruci imati jednu vilicu kad jede. Neke osobe glasno jedu što ometa neke druge osobe koje ne mogu istodobno jesti uz taj zvuk. Stoga osobe koje glasno jedu neće započeti jesti ako neka druga osoba kojoj to smeta već jede, ali i osoba kojoj smeta glasno jedenje neće početi jesti ako neka druga tada glasno jede (pričekati će da završi). Nakon što se najede osoba opere vilice i vraća ih na sredinu stola. Simulirati osobe dretvama i sinkronizirati ih monitorima. Pretpostaviti da su parametri za dretvu broj ruku (`br_ruku`), oznaka (0/1) da li osoba jede glasno (`glasno_jede`) te oznaka (0/1) smeta li osobi glasno hranjenje drugih osoba (`smeta_glasno`). Napisati početne vrijednosti korištenih varijabli. Nije potrebno rješavati problem izglednjivanja, ali ako postoji u rješenju navesti scenarij njegove pojave.

Rješenje

globalne varijable s početnim vrijednostima:

```
vilica_na_stolu = 20
```

```
br_glasno_jedu = 0
```

```
br_smeta_glasno = 0
```

```
dretva osoba(br_vilica, glasno_jede, smeta_glasno) //0/1 za *jede/*smeta
{
    ponavlja {
        razmišlja;

        udi_u_monitor(m)
        dok je (vilica_na_stolu < br_vilica) ||
            (glasno_jede I br_smeta_glasno) || (smeta_glasno I br_glasno_jedu)
            čekaj_u_redu_uvjeta(red, m)
        vilica_na_stolu -= br_vilica
        br_glasno_jedu += glasno_jede
        br_smeta_glasno += smeta_glasno
        izadi_iz_monitora(m)

        jede;

        udi_u_monitor(m)
        vilica_na_stolu += br_vilica
        br_glasno_jedu -= glasno_jede
        br_smeta_glasno -= smeta_glasno
        oslobodi_sve_iz_reda_uvjeta(red)
        izadi_iz_monitora(m)
    }
}
```

Zadatak

Poslovi P1, P2 i P3 se javljaju periodički svakih 200 ms. Unutar svake periode P1 se javlja 20 ms prije P2 koji se javlja 10 ms prije P3. Obrada za P1 traje 50 ms, za P2 30 ms te za P3 20 ms. Izračunati:

- a) prosječan broj dolazaka novih poslova u jedinici vremena
- b) prosječno trajanje obrade
- c) prosječan broj poslova u sustavu
- d) prosječno zadržavanje poslova u sustavu.

Rješenje

simulacija – jedan znak = 10 ms

2. u redu: 33
1. u redu: 222333
Poslužitelj:1111122233-----

a) $\alpha = ?$

tri posla svakih 200 ms $\Rightarrow \alpha = 3 / 200 \text{ ms} = 3/0,2 = 15 \text{ 1/s}$

b) $1/\beta = ?$

imamo zadana trajanja za pojedine poslove koji jednako dolaze (jednom u 200 ms) pa računamo prosjek

$$1/\beta = (50+30+20)/3 = 100/3 = 33,3 \text{ ms}$$

c) $\bar{T} = ?$

iz simulacije vidimo koliko se koji posao zadržao; računamo prosjek

$T_1=50 \text{ ms}$, $T_2=60 \text{ ms}$, $T_3=70 \text{ ms}$,

$$\bar{T} = (50+60+70)/3 = 180/3 = 60 \text{ ms}$$

d) $\bar{n} = ?$

prema Littleovoj formuli računamo:

$$\bar{n} = \alpha * \bar{T} = 15 * 0,06 = 0,9$$

Zadatak

Zbog preopterećenja nekog poslužitelja koji je radio s 90 % opterećenjem, trećina zahtjeva koje je on dobivao je preusmjerena drugdje. Ako je prije prosječno vrijeme zadržavanja zahtjeva u tom poslužitelju bilo 1 s, koliko će biti nakon ovog preusmjeravanja (samo za zahtjeve u ovom poslužitelju)?

Rješenje

$$\rho = 0.9$$

$$\alpha' = 2/3 \alpha$$

$$\bar{T} = 1$$

iz ρ i $\bar{T}$ kroz formule $\rho = \alpha/\beta$ i $\bar{T} = 1/(\beta - \alpha) \Rightarrow \alpha = 9, \beta = 10$ te onda $\alpha' = 6$

$$\bar{T}' = 1/(\beta - \alpha')$$

$$\bar{T}' = 1/(10 - 6) = 1/4 = 0,25 \text{ s}$$

Zadatak

U nekom sustavu se izvode dretve 1a, 1b, 2a, 2b, 3a, 3b. Sve se raspoređuju prema prioritetu kao primarnom kriteriju te podjelom vremena kao sekundarnom uz kvant $T_q=1$ (način SCHED_RR). Prioritet dretvi određen je njenim indeksom - 3a i 3b imaju najveći prioritet 3. Dretve se javljaju u sustavu u trenucima: $(t, D) = \{(0, 2a), (1, 1a), (2, 2b), (3, 3a), (5, 1b), (9, 3b)\}$. Pokazati izvođenje dretvi do $t=20$ ukoliko svaka dretva treba četiri jedinice vremena za svoje izvođenje. U trenucima kad se više događaja događa istovremeno proizvoljno odabrati redoslijed.

Rješenje

Pripravne dretve (nije potrebno pisati)																						
						1b	1b			1b	1b	1b	1b									
				1a	1a	1a	1a	1b	1b	1a	1a	1a	1a	1b								
			1a	2b	2b	2b	2b	1a	1a	2b	2b	2b	2b	1a	1b	1b						
		1a	2a	2a	2a	2a	2a	2b	2a	2a	2a	2a	2a	2b	1a	1a	1b	1a	1b	1a		
Aktivna	2a	2a	2b	3a	3a	3a	3a	2a	2b	3b	3b	3b	3b	2a	2b	2b	1a	1b	1a	1b		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Zadatak

Za neki proces zadana je tablica prevođenja (desno). Ako je veličina stranice 256 B koju će fizičku adresu generirati sklop za logičku adresu 0x0281?

str.	okvir	bit pris.
0	0x16	1
1	-	0
2	0x23	1
3	0x48	1
4	-	0

Rješenje

iz veličine stranice vidimo da treba 8 bita za odmak: zadnje dvije heksa znamenke se kopiraju, a one prije označavaju adresu stranice/okvira

0x0281 => stranica = 0x02, odmak = 0x81

stranica 0x02=2 je u okviru 0x23 (prema tablici)

pretvorba 0x0281 => 0x2381

Zadatak

U sustavu koji koristi dinamičko upravljanje spremnikom s 8 MB spremnika javljaju se zahtjevi za pokretanjem procesa te završetka prema redu: +P4, +P2, -P4, +P1, +P3, -P2, +P2, -P3, +P4, -P2, +P5. Plus ispred oznake procesa +PX označava zahtjev za pokretanje procesa PX koji treba X MB radnog spremnika, dok -PX označava kraj izvođenja tog procesa. Prikazati stanje spremnika za zadane zahtjeve sve do kraja ili do trenutka kada se zahtjev ne može poslužiti (nema dovoljno velike rupe).

Rješenje

zahtjev	radni spremnik							
+P4	4	4	4	4				
+P2	4	4	4	4	2	2		
-P4					2	2		
+P1					2	2	1	
+P3	3	3	3		2	2	1	
-P2	3	3	3				1	
+P2	3	3	3	2	2		1	
-P3				2	2		1	
+P4	nema mjesta							
-P2								
+P5								

Pri posluživanju zahtjeva se uvijek koristi najmanja rupa koja je dovoljno velika

Zadatak

Neki disk ima 4 površine od kojih svaka ima 512 staza, svaka staza ima 4096 sektora, a svaki sektor je velik 4 kB. Disk se okreće s 6000 okr/min. Disk radi na načelu da pročita cijelu stazu u interni spremnik te ju prenosi u radni spremnik – prijenos traje 5 ms. Prosječno postavljanje glave iznosi 10 ms, a premještanje na susjednu stazu 1 ms. Na tom disku kompaktno je pohranjena datoteka veličine 160 MB. Koliko će trajati čitanje cijele datoteke u radni spremnik (uz zanemarenje svih ostalih operacija, tj. pretpostavke da je opisnik datoteke već učitao u radni spremnik te da nema drugih zahtjeva prema disku)?

Rješenje

staza = 4096*4 kB = 16 MB

datoteka = 160 MB => 10 staza => 2 puna cilindra + 2 staze na trećem cilindru

6000/60 = 100 okr/s => $T_R=10$ ms, $T_R/2=5$ ms

$$t = T_{\text{seek}} + (T_R/2 + T_R + T_P) + (T_R/2 + T_R + T_P) + (T_R/2 + T_R + T_P) + (T_R/2 + T_R + \max(T_P, T_1)) + \\ (T_R/2 + T_R + T_P) + (T_R/2 + T_R + T_P) + (T_R/2 + T_R + T_P) + (T_R/2 + T_R + \max(T_P, T_1)) + \\ (T_R/2 + T_R + T_P) + (T_R/2 + T_R + T_P)$$

$$\max(T_P, T_1) = T_P$$

$$t = T_{\text{seek}} + 10 * (T_R/2 + T_R + T_P) = 10 + 10 * (5 + 10 + 5) = 10 + 10 * 20 = 210 \text{ ms}$$