

Operacinės sistemos

Galimi operacinių sistemų klausimai:

- 1) Semaforai.
- 2) Puslapiavimo mechanizmas.
- 3) Puslapiavimo strategijos.
- 4) Užduoties išlygiagretinimas.

1) **Užduotis.** Kokią reikšmę įgys bendrasis semaforas X užsiblokavus abiem procesams P1 ir P2:

SEMAPHORE R, S, X, T

R:=S:=X:=0; T:=2;

P1: BEGIN

```
    L1: V(X);
        P(R);
        V(S);
        GOTO L1;
END
```

AND

P2: BEGIN

```
    L2: V(X);
        V(R);
        P(S);
        P(T);
        GOTO L2;
END
```

ATS

Bendrasis semaforas yra toks, kuris gali įgyti bet kokią teigiamą reikšmę. Dvejatainis semaforas yra toks, kurio įgyjamų reikšmių aibė yra 0 ir 1. Primityvas $P(A)$, kai A bet koks semaforas, semaforo A reikšmę sumažina vienetu t.y $A=A-1$. Primityvas $V(A)$, padidina semaforo reikšmę vienetu t.y $A=A+1$. Jei semaforo reikšmė lygi 0 ir kviečiamas primityvas $P(A)$, tai procesas, kuris kviečia primityvą P, užsiblokuoja.

Šiuo atveju laikome kad procesų P1 ir P2 prioritetai vienodi. Vadinasi pirmiausia bus vykdomas tas, kuris eina pirmiau. Šiuo atveju tai P1. Kiekvienam procesui yra paskirtas laiko kvantas, kurį jis dirba, o po to grąžina procesorių. Laikome, kad laiko kvantas didesnis, nei procesorius dirbs t.y proceso darbas bus pertraukas tik tada, kai jis bandys iššaukti primityvą P(A), kai A reikšmė lygi 0 ir galės tęsti darbą tik tada, kai bus grąžintas procesorius ir $A > 0$.

Procesas	X	R	S	T
	0	0	0	2
P1	1	0	0	2
P2	2	0	0	2
P2	2	1	0	2
P1	2	0	0	2
P1	2	0	1	2
P1	3	0	1	2
P2	3	0	0	2
P2	3	0	0	1
P2	4	0	0	1
P2	4	1	0	1
P1	4	0	0	1
P1	4	0	1	1
P1	5	0	1	1
P2	5	0	0	1
P2	5	0	0	0
P2	6	0	0	0
P2	6	1	0	0
P1	6	0	0	0
P1	6	0	1	0
P1	7	0	1	0

Procesas P1 kaskart užsiblokuoja ties primityvu P(R). Tokiu atveju procesorius perduodamas procesui P2, o pastarasis užsiblokuoja ties primityvu P(S). Šitaip veikdami abu procesai niekada neužsiblokuotų kartu, jei ne semaforas T. Kadangi semaforui T kviečiamas tik P (T) primityvas,

tai po tam tikro laiko susidaro situacija kai P1 procesas negali vyksti toliau nes $R=0$, o P2 negali vyksti, nes $T=0$. Tokiu atveju gaunama aklavietė.

Ats.: bendrojo semaforo reikšmė lygi 7.

2) **Užduotis.** PTR registro reikšmė lygi 5294. Puslapių lentelė užpildyta reikšmėmis nuo 9 iki 0. Vykdoma komanda PD28. Rasti absoliutų (realų) adresą.

Šiuo atveju efektyvus (lokalus) adresas yra 28. Į komandos kodą PD galime nekreipti dėmesio. PTR registrą sudaro 4 skaitmenys, tačiau naudojami tik du paskutiniai.

a) Pirmiausia reikia apskaičiuoti bloko numerį. Skaičiavimas vyksta taip: $9 \cdot 10 + 4 = 94$. Kitaip sakant du paskutiniai PTR skaitmenys nurodo bloko numerį.

b) Reikia susinumeruoti puslapių lentelę. Norinėdami tai padaryti, turime rasti pirmą indeksą. Jis randamas padauginus bloko numerį iš dešimties: $94 \cdot 10 = 940$.

Susinumeruojame:

940	941	942	943	944	945	946	947	948	949
9	8	7	6	5	4	3	2	1	0

c) Randame žodį: (Bloko nr) $\times 10 +$ (EA pirmas skaitmuo) $= 94 \cdot 10 + 2 = 942$ (primename kad šiuo atveju EA=28, o pirmas skaitmuo yra 2).

d) Randame AA. Kai jau turime žodį 942 lieka apskaičiuoti AA. Tai darome tokiu būtu: imame žodžio reikšmę PSL lentelėje. Šiuo atveju tai bus 7. Dauginame iš 10 ir pridedame poslinkį, kuris yra EA paskutinis skaitmuo. $7 \cdot 10 + 8 = 78$.

Ats.: AA=78.

3) **Puslapiavimo strategijos.** Virtuali atmintis (VA) dažniausiai buna didesnė už realią atmintį (RA). Dėl šios priežasties kai pritrūksta virtualiai mašinai (VM) VA, tenka išsisaugoti puslapio reikšmę išorinėje atmintyje ir jo vietoje pakrauti kitą puslapį. Apsikeitimas su išorine atmintimi yra gana lėtas, todėl kuo mažiau kartų reikės kreiptis į išorinę atmintį (keisti puslapius) tuo geriau. Kad realizuoti tokią puslapių apsikeitimo idėją buvo sugalvotos kelios puslapių apsikeitimo strategijos, kurias toliau ir aptarsime.

Užduotis. Parinkti optimalią puslapiavimo strategiją ir apskaičiuoti puslapių keitimo skaičių, kai duota virtualių puslapių trąsa: V1, V2, V3, V1, V4, V2, V3, V1 ir du realūs puslapiai B1 ir B2.

A) Biladi strategija. Jeigu reikia keisti puslapį, keisime tą, kuris trąsoje bus sutinkamas vėliausiai.

B) Keisti tą, kuris atmintyje buvo ilgiausiai.

C) Keisti tą, į kurį buvo kreiptasi vėliausiai (vėliausiai=paskutinis kreipinys).

A		B		C	
B1	B2	B1	B2	B1	B2
V1		V1		V1	
V1	V2	V1	V2	V1	V2
V1	V3	V3	V2	V1	V3
V1	V3	V3	V1	V1	V3
V4	V3	V4	V1	V4	V3
V2	V3	V4	V2	V2	V3
V2	V3	V3	V2	V2	V3
V2	V1	V3	V1	V2	V1
KEITIMŲ SKAIČIUS: 4		KEITIMŲ SKAIČIUS: 6		KEITIMŲ SKAIČIUS: 4	

4) Užduoties išlygiagretinimas. Tarkim turime algebrinių veiksmų eilutę. Ją galime išskaidyti į atskiras dalis, kurias atliksime lygiagrečiai ir nuosekliai. Veiksmus, kuriuos galėsime atlikti lygiagrečiai žymėsime „and“, o kuriuos nuosekliai „;“; “. Norint išlygiagretinti veiksmus, galimi keli vykdymo variantai. Vieną iš jų ir pateiksime toliau.

Užduotis. Parašyti algoritmą pseudokodu, kuris išlygiagretina algebrinius veiksmus šios eilutės: $(a+b+c*b)+(a*b+c)/(b+c*d);$

BEGIN

T1:=c*b; T2:=T1+a+b;

END

AND

BEGIN

T3:=a*b; T4:=T3+c **AND**

$T5:=c*d$; $T6:=T5+b$;

$T7:=T4/T6$; $T8:=T2+T7$;

END

Negali išlygiagretinti tų dalių, kurios skaičiuojamos konvejeriniu principu t.y kad po to einančioje dalyje naudojamas prieš tai gautas rezultatas. Iš principo čia buvo galima daryti ir 3 blokus (blokas prasideda BEGIN ir baigiasi END), ir kiekvieno viduje skaičiuoti skliausteliais apskliaustas išraiškas ir paskutiniam blokui paskirti nuosekliai atlikti veiksmus, kurie sieja išraiškas.

SĖKMĖS RUOŠIANTIS EGZAMINUI ☺