

SUBIECTE APP

Subiect B 2024

1. Ce tip de parallelism nu toleră sistemele runtime din cadrul mediulor de programare parallele?

CURS 5

RTS = extenție funcțională a OS în user space

Tipuri de parallelism:

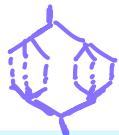
1) Flat Parallelism

= fără relație parinte/copil între thread-uri

→ eficient pt. pb. independ., dar greu de fol. pt. app. cu echilibru între thread-uri

2) Nested Parallelism

(parallelism hierarhic)



→ thread-urile pot spawnă alte thread-uri

→ bun pt. sist. heterogene, structuri de date complexe

→ fol. de OpenMP



3) Dynamic Parallelism

→ thread-uri create dinamic când avem parallelism variabil

→ bun load balancing pt. app. dinamice

→ RĂSPUNS: NU e parallelism implicit

Parallelism implicit = user-ul sunăWARE de parallelism, e gestionat de biblioteci
= single-threaded view

2. MIC₁...5

$\mu\text{IC}_1 = f$	d	a	c	b	g	h	j	k	l
$\mu\text{IC}_2 = h$	d	a					x	x	
$\mu\text{IC}_3 = j$	e	d	c				x	x	
$\mu\text{IC}_4 = q$	k	d	c	b					
$\mu\text{IC}_5 = q$	k	j	h	g	f				

Organizare optimă cîmpuri din formatul general al microimpr.=?

CURS 13

Algoritm de minimizare a nr. fizicii necesare pt. codificarea reprezentării complete.

1) Tabel initial:

	a	b	c	d	e	f	g	h	j	k	l
a	x								x		
b		xx	x								x
c		xx	x	x				x	x		
d			xx	xx	xx			x	x	x	
e				xx	x				x		
f					x		xx	xx	x	x	
g							xx	x	x	x	
h	x					x	xx	x	x	x	
i					xx	x	xx	x	x	x	
j						x	xx	x	x	x	
k							xx	x	x	x	
l								xx	x	x	

Ne uităm la locuri
Nici două sunt
în același rând.
le marcam cu
label (înseamnă
că sunt microgo.
incompatibile)

2) Clase maxime incompatibilitate:

$\text{MIC}_1 = \{a, d\}$

$\text{MIC}_2 = \{a, h\}$

$\text{MIC}_3 = \{b, c, d, k, l\}$

$\text{MIC}_4 = \{c, d, j, k, l\}$

$\text{MIC}_5 = \{c, e, j\}$

$\text{MIC}_6 = \{d, e, j\}$

$\text{MIC}_7 = \{f, g, h, j, k\}$

3) Clase maxime compatibilitate:

$\text{MCC}_1 = \{a, b, e, f\}$

$\text{MCC}_2 = \{a, c, f\}$

$\text{MCC}_3 = \{a, e, g\}$

$\text{MCC}_4 = \{a, e, k\}$

$\text{MCC}_5 = \{a, e, l\}$

$\text{MCC}_6 = \{b, e, g\}$

$\text{MCC}_7 = \{b, e, h\}$

$\text{MCC}_8 = \{b, j\}$

$\text{MCC}_9 = \{c, g\}$

$\text{MCC}_{10} = \{c, h\}$

$\text{MCC}_{11} = \{d, g\}$

Sol. simplă: Reap. final b. să includă cîmpuri parte din MCC-uri.

Verificăm grile.

Camp1	Camp2	Camp3	Camp4	Camp5	Camp6	Camp7
(q)	(k)	(j)	(h e b)	★ (g d)	(f c a)	★ (d)
(q)	(k)	(j)	(h c)	★ (g a)	(f e b)	★ (d)
(q)	(k)	(j)	(h f d c)	X (g e a)	(b)	
(q)	(k)	(j)	(h e c b a)	X (g)	(f)	(d)

a, h incomp.

f, l incomp.

⇒ Avem var. a și b. Dintre acestea, alegem varianta cu mai puține cîmpuri (optimă): A

Continuare algoritmul:

3) Tabelul de acoperire modif.:

• Dezm. MIC_3 clasa incomp. max.

• $\text{MIC}_3 = \{b, c, d, k, l\}$

$b \subseteq \{\text{MCC}_1, \text{MCC}_6, \text{MCC}_7, \text{MCC}_8\}$

$c \subseteq \{\text{MCC}_2, \text{MCC}_9, \text{MCC}_{10}\}$

$d \subseteq \{\text{MCC}_1\}$

$k \subseteq \{\text{MCC}_4\}$

$l \subseteq \{\text{MCC}_5\}$

• $\text{AMCC}_3 = \{\text{MCC}_1, \text{MCC}_2, \text{MCC}_4, \text{MCC}_5, \text{MCC}_6, \text{MCC}_7, \text{MCC}_8, \text{MCC}_9, \text{MCC}_{10}, \text{MCC}_{11}\}$

	a	b	c	d	e	f	g	h	j	k	l
MCC_1	X	X									
MCC_2	X										
MCC_4	X										
MCC_5	X										
MCC_6		X									
MCC_7		X									
MCC_8		X									
MCC_9			X								
MCC_{10}			X								
MCC_{11}				X							

$\text{MCC}_4, 5, 8, 11$ esentiale

4) Tabella de acoperire redusa

• excludem op. incluse în MCC esențiale și
MIC max.:

$$MIC_3 = \{b, c, d, k, g\}$$

$$MCC_1 = \{a, e, k\}$$

$$MCC_5 = \{a, e, g\}$$

$$MCC_8 = \{b, j\}$$

$$MCC_{11} = \{d, g\}$$

\Rightarrow Rămân f, h.

	f	h
MCC ₁	X	
MCC ₂	X	
MCC ₇		X
MCC ₁₀		X

$$\Rightarrow \left\{ \begin{array}{l} MCC_1, MCC_7 \\ MCC_1, MCC_{10} \\ MCC_2, MCC_2 \\ MCC_2, MCC_{10} \end{array} \right.$$

P

5) Set soluții:

SOL = clase esențiale + ...

$$= \{ MCC_{4,5,8,11,1,7}, MCC_{4,5,8,12,1,10}, \\ MCC_{4,5,8,12,2,7}, MCC_{4,5,8,12,2,10} \}$$

?? m-am pierdut
mai în me faceam

3. Diferență Data Flow static / dinamic

CURS 3

Data Flow

→ model de calcul paralel în care execuția unei cicluri de prelucrare e efectuată deoarece dacă și operanțul

→ nu ca PC, activarea se face când sunt disp. toți operații

Data Flow Static

→ iterările consecutive ale unei bucle se execută în pipeline

→ se fol. unde dep. de date sunt stabilite la compilare

→ bun când avem program simplu, model mai puțin flexibil

Data Flow Dinamic

→ fiecare iteratie buclă / invocare subprogram se poate executa într-o instanță separată de subgraf

→ când dep. date sunt stabilite la runtime

→ bun pt. program complex, model mai flexibil

Caracteristica	Static Data Flow	Dynamic Data Flow
Dependenta date	Determinat la compilare	Determinat la runtime
Flexibilitate	Puțin flexibil	Mai flexibil
Complexitate	Puțin complex	Mai complex

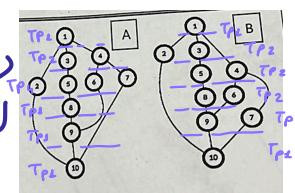
5. $P = 12\ 000$ proceseare

JOB $\rightarrow 10$ sarcini ($N = 10$)

$t = 10$ ms (sequential)

2 grafuri dep. A, B

$f_{seq} = 5\%$, $f_{par} = 95\%$



CURS 3

1) Care din grafuri de dep. conduc la o execuție mai rapidă? Justificați calculând V_A, V_B

2) Care graf e mai eficient (economic)?

Calculăm timpul serial:

$$(A) T_1 = N \cdot t = 100 \quad \leftarrow \text{timp execuție graf serial lucrat în II}$$

$$T_{P1} = f_{seq} t + \frac{f_{par} \cdot t}{P}$$

$$T_{P2} = f_{seq} t + \frac{f_{par} \cdot t}{P/2}$$

$$T_{P4} = f_{seq} t + \frac{f_{par} \cdot t}{P/4}$$

$$T_P = 4 T_{P1} + T_{P2} + T_{P4} =$$

$$= 6 \cdot f_{seq} \cdot t + \frac{4 f_{par} \cdot t}{P} + \frac{f_{par} \cdot t}{P} + \dots + \frac{f_{par} \cdot t}{P/4} =$$

luăm în II
fiecare nivel graf
punem joi P supra
cîte noduri pe
nivel (n'imparam
egal întreanul)

$$= 6 \cdot 0,05 \cdot 10 + \frac{4 \cdot 0,95 \cdot 10}{12 \cdot 10^3} + \frac{0,95 \cdot 10}{6 \cdot 10^3} + \frac{0,95 \cdot 10}{3 \cdot 10^3}$$

$$= 3 + \frac{38 + 19 + 38}{12 \cdot 10^3} = 3 + 0,0071 = 3,0071$$

$$V_p = \frac{T_L}{T_p} = \frac{100}{3,0071} = 33,2546$$

$$\textcircled{B} \quad T_p = 2T_{p1} + 4T_{p2} =$$

$$= 6 \cdot 0,05 \cdot 10 + 2 \cdot \frac{0,95 \cdot 10}{12 \cdot 10^3} + 4 \cdot \frac{0,95 \cdot 10}{6 \cdot 10^3} =$$

$$= 3 + \frac{9,5 + 38}{6 \cdot 10^3} = 3 + 0,0079 = 3,0079$$

$$V_p = \frac{T_L}{T_p} = \frac{100}{3,0079} = 33,2457$$

$V_A > V_B \Rightarrow$ se exec. mai rep. A

6. Ce pas nu este necesar pentru a imbunatati rularea codurilor paralele ce scaleaza slab sau puternic?
- Evitati zonele secentiale de cod
 - Utilizati algoritmi eficienti de comunicare
 - Echilibrati incarcarea workloadurilor intre procesoare
 - Utilizati algoritmi eficienti de load-balancing
 - Eliminati sincronizările din cod

Strong Scaling din start e asta ptc.
sincronizările sunt nevoie
cum timpul de exec. creste,
odată ce nr. proc. creste (dim. prob. fixă)

- măsoară cum timpul de exec. creste,
- deci e imp. scăderea timpului total.
- ⇒ num strong scaling

- util pt. granularitate crescută
- măsoară parallelism intrinsic
- deci e imp. scăderea timpului total.
- ⇒ num strong scaling

Weak Scaling

- măsoară cum timpul de exec. creste, odată cu dim. problemei și cu nr. de proc., dn timp ce incarcarea pe proc. rămâne ct.
- util pt. dim. crescute
- măsoară scalabilitate
- deci num rez. probleme mari (cu nr. proc. mare) ⇒ num weak scaling

Strong/Weak Scaling Powerups

- Evitati zonele secentiale de cod
- Folositi algoritmi eficienti de comunicare
- Utilizati algoritmi eficienti de load-balancing
- Echilibrati incarcarea workloadurilor intre procesoare
- Realizati un pas de tunare / optimizare al executiei aplicatiilor paralele pentru hardware-ul specific pe care se realizeaza rularea

Sincronizări → esențiale în multe cauzi pt. a asigura comunicare corectă

7. $C = (S, \leq)$

$S = \{A, B, C, D, E, F, G, H\}$

Să se construiască $C' = (S', \leq')$ echivalent, care realizează max. de paralelism, înințând cont de cerințele de determinare. $NV = ?$ nr. proc. necesari?

M	Domeniu de definiție D _{sarcina}	Domeniu de valori R _{sarcina}
M1	A,B,G,H	C
M2	A,G	E
M3	C,D,H	A
M4	C,D,E,G	B,G
M5	F	D,F,H

CURS 9

$$Z = \{(S, S') \in \leq \mid (R_S \cap R_{S'}) \cup (R_S \cap D_{S'}) \cup (D_S \cap R_{S'}) \neq \emptyset\}$$

Relația Z, conform definiției de mai sus va fi dată de mulțimea arcelor:

Z : {	(S1,S2) (S1,S4) (S1,S5) (S1,S8) (S2,S3) (S2,S4) (S2,S5) (S2,S7) (S3,S7) (S3,S8) (S4,S6) (S4,S7) (S4,S8) (S5,S7) (S6,S8)	D _S	R _S
M1	(S1,S2,S7,S8)	S1,S7	S5
M2	S1,S7	S5	
M3	S3,S4,S5,S7	S1	
M4	S3,S4,S5,S7	S2,7	
M5	S6	S4,S6,S8	

$D_S \cap R_{S3} \neq \emptyset$

Rezultatul grafic asociat G', care este organizat pe niveluri eliminând arcele exclude prin tranzitivitate

$D_S \cap R_S \neq \emptyset$

Explanare: Vrem să luăm toate percheile de sarcini care vor să serie (R_S) și să citească (D_S) din aceeași niv.
⇒ aceste sarcini nu se pot executa în împreună.
⇒ vom adăuga un arc de dependență în grafic de dep.

$$Z = \{(A,C), (A,D), (A,E), (A,H), (B,C), (B,D), (B,E), (B,G), (C,G), (C,H), (D,G), (D,F), (D,H), (E,G), (F,H)\}$$

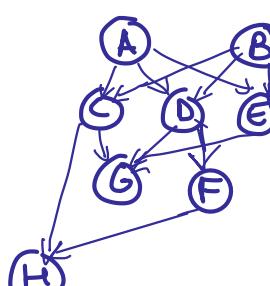
Eliminăm arcele excluse prin hensitivitate:

$\{A,C\} \rightarrow$ eliminăm (A,H)

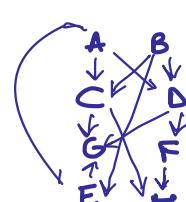
$\{B,C\} \rightarrow$ eliminăm (B,G)

$\{D,F\} \rightarrow$ eliminăm (D,H)

Rămân celelalte arce din Z și constituim graf.



dacă merge și pe 2 proc.:



3 procesare ⇒ 6 min.

2 proc. ⇒ 4 min.

Considerând starea curentă a unui sistem de sarcini $S = \{S_0, S_1, S_2, S_3, S_4\}$ care dispune de resursele $R = A_1, A_2, A_3, A_4$

Alocare maximă disponibilă

	Alocarea curentă				Maxim necesar				Disponibile			
	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4
S0	0	0	1	2	0	0	1	2	1	5	2	0
S1	1	0	0	0	1	7	5	0				
S2	1	3	5	4	2	3	5	6				
S3	0	6	3	2	0	6	5	2				
S4	0	0	1	4	0	6	5	6				

Răspundeți la următoarele întrebări:

- Care este conținutul resurselor de care este nevoie pentru executarea sistemului de sarcini?
- Sistemul este într-o stare sigură?
- Dacă se poate o cerere de la sarcina S1 pentru resursele (0, 4, 2, 0), poate solicitarea să fie acordată imediat?

Justificati

CURS 10

a) Necessar pt. finalizare = Maxim necesar - alocare curentă

	A1	A2	A3	A4
S0	0	0	0	0
S1	0	7	5	0
S2	1	0	0	2
S3	0	0	2	0
S4	0	6	4	2

b) Disponibile: $(1, 5, 2, 0)$ după S_0

\Rightarrow Se elibereză S_0 $\overline{(0, 0, 0, 0)}$ $\oplus (1, 5, 2, 0)$

\Rightarrow Se elibereză S_1 $\overline{(0, 7, 5, 0)}$ $\oplus (2, 5, 3, 2)$

\Rightarrow Se elib. S_2 $\overline{(1, 3, 5, 1)}$ $\oplus (3, 8, 8, 6)$

\Rightarrow Se elib. S_3 $\overline{(1, 3, 5, 1)}$ $\oplus (3, 8, 8, 6)$

TO DO

9. Concept weak scaling + exemplu CURS 3

Weak Scaling

→ măsoară cum timpul de exec. crește, odată cu dim. problemei și cu nr. de proc., dn timp ce concurența pe proc. rămâne ct.

→ util pt. dim. crescute

→ măsoară scalabilitate

→ dacă vom rez. probleme mari (cu nr. proc. mare) \Rightarrow vom weak scaling

Exemplu: Avem un cod paralel

• Dim = 1 M Proc = 1 Timp = 1000 s	• Dim = 2 M Proc = 2 Timp = 1000 s
--	--

\Rightarrow Weak scaling perfect.

Programul scalează la dim. mari fără a pierde performanță.

10. CURS 10

		Descrieti in pseudocode Pri, precizati ce operatii sunt indivizibile	Descrieti in pseudocode Uri, precizati ce operatii sunt indivizibile
10	Pri Si Uri		

Algoritm de excludere mutuală din mediu centralizat

$Pri_i \rightarrow$ precede imediat sarcina Si

\rightarrow Pseudocod:

$Q[p] = i$
 $p = i$] punem pe Si în coadă

dacă $Q[0] = i \rightarrow$ cand Si ajunge

do Pri_i capăt listei,
altfel wait facem bl;

(ex: $Q[3] = 2 \rightarrow$ procesul 2 e al 3lea din coadă)

\rightarrow operații indivizibile: $Q[p] = i, p = i$

$Uri_i \rightarrow$ urmează imediat sarcină Si

\rightarrow Pseudocod:

dacă $p = i$ \leftarrow dacă Si a fost ultima sarcină din coadă

$p = 0$] face coada vidă

$Q[0] = 0$ altfel

$Q[0] = Q[i]$ altfel, stace Si la execuție

cum capăt listei;

\rightarrow op. indivizibile: teste ($p = 0, Q[0] = 0, Q[0] = Q[i]$)

Subiect A

CURS 4

1. P.p. că avem o structură cu p proc. pe care se execută în secvenție, avem timp sincron.
 ? t₀ și timp overhead t₀. Cătăram m. proc.
 f. mult și m. secvenție rămâne ct.
 Care dimine param. afectează cel mai mult viteza de execuție? (t₀ / t₀ / [m/p]).

Limita lui Worton: (se fol. formula dim cursul de la NT?)

$$V_p = \frac{N \cdot t}{t_0 + [N/p] \cdot (k + t_0)}$$

↑ const.

$$p \rightarrow \infty \Rightarrow [N/p] \rightarrow 0 \rightarrow V_p = \frac{N \cdot t}{t_0}$$

=) ↑ const.

? Dacă N → ∞ ⇒ t₀ const.

2. Structură el. proc. slotică conectată în pipeline linear-unidimensional

? Câte el. procesare pt. efectuare

$$A[m, m] \cdot B[m]$$

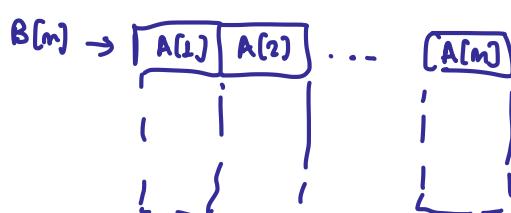
CURS 8

Timp execuție? (timp exec. pe el. de execuție = t_c, constant)

Slotic linear → căte num PE pt. fiecare linie din A[m, m]

⇒ avem nevoie de m procesare, unul pt. fiecare linie a lui A

→ B e introdus stepat în pipeline



$$\begin{aligned} \text{Timp} &= m \text{ cicluri complete PE-uni cu } B[i] \\ &\quad m-1 \text{ cicluri pt. urm. golii} \\ &= 1 \underline{m+m-1} = 2m-1 \end{aligned}$$

3.

Care este timpul de execuție pentru adunarea unor vectori C[n]=A[n]+B[n], elementele fiind reprezentate în virgula mobilă. Considerăm n=100 numărul de elemente și timpul mediu de efectuare a unei operații elementare în UAL, tmed=10 microsec.

Intr-o structură monoprocesor cu UAL fără pipeline

Intr-o structură monoprocesor cu UAL cu structură pipeline cu 4 elemente de execuție (Comparare caracteristici, Deplasare mantisa, dacă este cazul, Adunare mantisa, Normalizare rezultat)

Intr-o structură SIMD cu n procesoare și UAL fără pipeline

a) $T = N \cdot t = 100 \cdot 10 = 1000 \mu s = 1 ms$

b) pipeline și el. exec.:

$$T = \left(\frac{m}{4} + 3 \right) \cdot t_{med} = 0,28 ms$$

c) SIMD cu m proc. SIMD → se exec. inst. din 1 pe N proc.

$$T = \frac{N \cdot t}{N} = t = 0,10 ms$$

4. Explicații XCHG

CURS 10

XCHG (n, m) → interchimbă ce o op. atomică conținutul registrelui n cu celula de memorie m (semifor)

n_i = variabilă locală a sarcinii S;
 ↑ initial 0

m = variabilă globală, comună tuturor sarcinilor
 ↑ initial 1

n_i, m pot lua valoare 0 sau 1

Algoritm excludere mutuale:

<început sarcină>; dacă proc. găsește m=1, obține acces la secțiunea critică și m devine 0.

XCHG (n_i, m); n_i devine 1 ⇒ proc. poate intra în sec. critică.

până când n_i; altă proc. rămâne în buclă,

<secțiune critică>; deosebe m=0

XCHG (n_i, m); după term. op. critică,
 se rezetează m=1

<rest sarcină>; și n_i=0.

5. (A, B, C, D) clase compatibilitate

$$|PO| = (11, 8, 10, 3)$$

A	B	C	D
---	---	---	---

CURS 11

Lungime cur. din biți a MC pt.:

a) Codificare verticală:

→ mixtă în 11, curv. scurt

A	B	C	D
---	---	---	---

de nu adăugăm aici bit activare bloc?

$$\rightarrow L = \log_2 (11 + 8 + 10 + 3) = \log_2 32 = 5 \text{ biți}$$

b) codificare orizontală

→ toate în 11, curv. lung

$$N_{A_1} \dots N_{A_{11}} N_{B_1} \dots N_{B_8} \dots N_{C_2} \dots N_{C_3}$$

$$\rightarrow L = 11 + 8 + 10 + 3 = 32 \text{ biți}$$

c) codificare minimă

→ mixtă pe cămpuri = codif. oriz.

→ în c.cmp = codif. verticală

→ clasele se codifică vertical:

$$A: L_A = \log_2 (11+1) = \log_2 12 = 4$$

$$B: L_B = \log_2 (8+1) = 4 \text{ bit activare bloc}$$

$$C: L_C = \log_2 (10+1) = 4$$

$$D: L_D = \log_2 (3+1) = 2$$

→ între clase codif. orizontale!

$$L = L_A + L_B + L_C + L_D = 4 + 4 + 4 + 2 = 14 \text{ biti}$$

6.

Fie un procesor care implementează structura paralela pipeline de citire interpretare execuție pentru procesare suprascală, cu patru unități paralele.
Presupunând un ciclu de instrucțiuni conține:
 -citire care necesită o singură perioadă de ceas
 -decodificare instrucțiunii necesită două perioade de ceas
 -execuția instrucțiunii necesită două perioade de ceas

 Avem o sevență de 3,000,000 instrucțiuni masina, iar frevența ceasului 2 Ghz
Calculati durata de executie a sevenței de program in milisecunde

Pipeline 4 unități paralele CURS 3

Ciclu: F (1 ceas), D (2 ceasuri), E (2 c)

$$m = 3 \cdot 10^6 \text{ instr. masina}$$

$$\tau = 2 \cdot 10^9 \text{ GHz}$$

$$t = ? \text{ ms}$$

F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂

după citire și decodificare,
putem executa în 1 ceas
facem urm. citire și
decodificare

F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂

F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂
F	D ₁	D ₂	E ₁	E ₂

?

Nr. cicluri:

$$N = 1 + 4 + \frac{3 \cdot 10^6}{4} \cdot 2 = 1,500,005 \cdot 10^6$$

$$T = \frac{N}{\tau} = \frac{1,500,005 \cdot 10^6}{2 \cdot 10^9} \approx 0,75 \text{ ms}$$

7.

CURS 13

Consideram microinstrucțiunile complete, μIC1... μIC5 care contin micro-operările ca în tabelul alăturat:

μIC1 = μo1 μo2 μo3 μo4 μo5 μo6

μIC2 = μo3 μo7 μo8 μo9

μIC3 = μo1 μo2 μo8 μo9 μo10

μIC4 = μo4 μo8 μo11

μIC5 = μo6 μo8

Care este organizarea optimă a campurilor din formatul general al microinstrucțiunilor:

	Camp1	Camp2	Camp3	Camp4	Camp5	Camp6	Camp7
a.	(μo1)	(μo2)	(μo3)	(μo4)	(μo5 μo9 μo11)	(μo6 μo7 μo10)	(μo8)
b.	(μo1)	(μo2)	(μo3)	(μo4 μo6)	(μo5 μo11)	(μo6 μo7 μo10)	(μo8)
c.	(μo1)	(μo2)	(μo3)	(μo4 μo9 μo10)	(μo5 μo11)	(μo6 μo7)	(μo8)
d.	(μo1)	(μo2)	(μo3)	(μo4 μo7 μo9 μo10 μo11)	(μo5)	(μo6)	(μo8)

	N01	N02	N03	N04	N05	N06	N07	N08	N09	N010	N011
N01	X	X	X	X	X	X	X	X	X	X	
N02	X	X	X	X	X	X	X	X	X	X	
N03	X	X	X	X	X	X	X	X	X	X	
N04	X	X	X	X	X	X	X			X	
N05	X	X	X	X	X	X					
N06	X	X	X	X	X	X	X				
N07							X	X	X		
N08	X	X	X	X	X	X	X	X	X	X	
N09	X	X	X				X	X	X	X	
N010	X	X					X	X	X		
N011				X			X			X	

$$1) \{N05 \ N09 \ N011\} \checkmark$$

$$\{N06 \ N07 \ N010\} \checkmark$$

$$2) \{N04 \ N09 \ N03\} \checkmark$$

$$\{N05, N011\} \checkmark$$

$$\{N06 \ N07 \ N010\} \checkmark$$

$$3) \{N04 \ N09 \ N010\} \times 9,10$$

$$4) \{N04 \ N07 \ N09 \ N010 \ N011\} \times 7,9$$

Dintre 1 și 2, deoarece avem același nr. de compun. vom să vedem care are lung. cel mai mic.

$$\begin{aligned} L_1 &= \log_2(1+1) + \log_2(1+1) + \log_2(1+1) + \\ &\quad \log_2(1+1) + \log_2(3+1) + \log_2(3+1) + \\ &\quad \log_2(1+1) = \\ &= 5 + 2 \cdot 2 = 9 \text{ biti} \end{aligned}$$

$$L_2 = 1 + 1 + 1 + 2 + 2 + 2 = 7 \text{ biti}$$

= varianta 1.

CURS 10

Fie un sistem de sarcini S=[S1, S2, S3, S4, S5] și patru resurse (R1, R2, R3, R4). Nu există nicio coadă de așteptare neacoperită de solicitări.

Resurse disponibile în prezent

Sarcina	R1	R2	R3	R4	Alocare curentă				Necesar Maxim				Necesar pentru finalizare			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
S1	0	0	1	2	0	0	3	2	0	0	2	0				
S2	2	0	0	0	2	7	5	0	0	7	5	0				
S3	0	0	3	4	6	6	5	6	6	6	2	2				
S4	2	3	5	4	4	3	5	6	2	0	0	2				
S5	0	3	3	2	0	6	5	2	0	3	2	0				

Justificați de ce este blocat sau de ce nu este blocat sistemul de darcini?

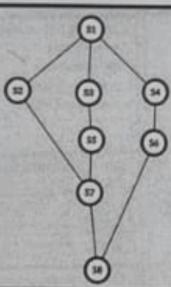
Dacă considerați că sistemul de sarcini nu este blocat, specificați o secvență de execuție posibilă.

o problemă pt. mâine

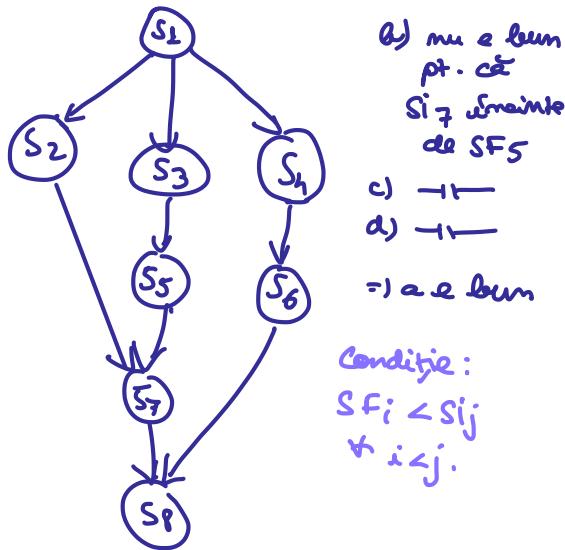


9.

Fiind dat sistemul de sarcini, care secvență de execuție este validă?



a)	S1	SF1	S12	S13	S14	SF3	S15	SF4	S16	SF2	SF5	S17	SF7	SF6	S18	SF8
b)	S1	SF1	S12	S13	SF3	S15	S17	SF2	S14	SF4	SF5	SF7	S16	SF6	S18	SF8
c)	S1	SF1	S12	S13	SF3	S15	S17	SF2	SF5	S14	SF4	S16	SF7	SF6	S18	SF8
d)	S1	SF1	S12	SF2	S13	SF3	S14	SF4	S15	S17	S16	SF6	SF5	S18	SF8	



10.

Un sistem multiprocesor dispune de n procesoare :

240 procesoare

Un JOB poate fi împărțit într-un număr de sarcini:

10 sarcini

Durata de execuție secvențială a unei sarcini este de:

10 sec $\approx 10^3$ mca

Dependența de date între sarcini este data în figura alăturată.

Considerând că sarcinile au o zonă de secvențialitate în procent de

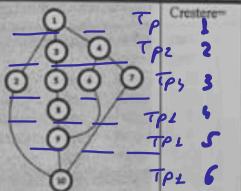
5% sarcină.

Să se calculeze creșterea de viteză a executării jobului pe structura multiprocesor:

5% procen

secv.

$$\sum f_{par} = 95$$



$$T = N \cdot t = 240 \cdot 10 = 2400$$

$$T_p = f_{seq} \cdot t + \frac{f_{par} \cdot t}{P}$$

$$T_{p2} = f_{seq} \cdot t + \frac{f_{par} \cdot t}{P/2}$$

$$T_{p4} = f_{seq} \cdot t + \frac{f_{par} \cdot t}{P/4}$$

$$T_p = 4 T_{p2} + T_{p4} + T_{p2} =$$

= ... - .