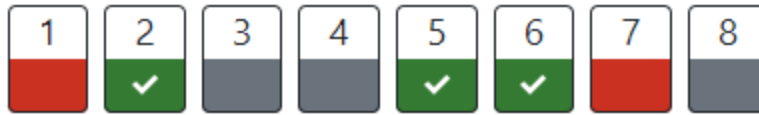


Quiz navigation



1. Considerând că în urma analizei microoperațiilor dintr-un subloc au rezultat microinstrucțiunile complete, mIC1,...mIC5 care conțin micro-operațiile ca în tabelul alăturat:

mIC1 = mo1 mo2 mo3 mo4 mo5 mo6

mIC2 = mo3 mo7 mo8 mo9

mIC3 = mo1 mo2 mo8 mo9 mo10

mIC4 = mo4 mo8 mo11

mIC5 = mo6 mo8

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



a.

Câmp 1: (mo1)

Câmp 2: (mo2)

Câmp 3: (mo3)

Câmp 4: (mo4 mo9)

Câmp 5: (mo5 mo11)

Câmp 6: (mo6 mo7 mo10)

Câmp 7: (mo8)



b.

Câmp 1: (mo1)

Câmp 2: (mo2)

Câmp 3: (mo3)

Câmp 4: (mo4 mo7 mo9 mo10 mo11)

Câmp 5: (mo5)

Câmp 6: (mo6)

Câmp 7: (mo8)



c.

Câmp 1: (mo1)

Câmp 2: (mo2)

Câmp 3: (mo3)

Câmp 4: (mo4 mo7 mo10)

Câmp 5: (mo5 mo8)

Câmp 6: (mo6 mo9 mo11)

Câmp 7:



d.

Câmp 1: (mo1)

Câmp 2: (mo2)

Câmp 3: (mo3)

Câmp 4: (mo4 mo9 mo10)

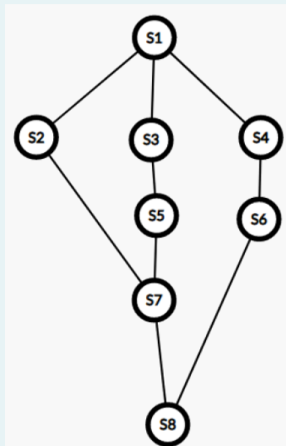
Câmp 5: (mo5 mo11)

Câmp 6: (mo6 mo7)

Câmp 7: (mo8)

Question **2**
Correct
Mark: 1.00 out of 1.00
Flag question

Fiind dat sistemul de sarcini



Care secvență de execuție este validă ?

- ☐ a. S11 SF1 SI2 SI3 SF3 SI4 SF4 SI5 SI7 SF2 SF5 SI6 SF6 SF7 SI8 SF8
- ☐ b. S11 SF1 SI2 SI3 SF3 SI5 SI7 SF2 SI4 SF4 SF5 SF7 SI6 SF6 SI8 SF8
- ☐ c. S11 SF1 SI2 SI3 SF3 SI5 SI7 SF2 SF5 SI4 SF4 SI6 SF7 SF6 SI8 SF8
- ☒ d. S11 SF1 SI2 SI3 SF3 SI5 SF5 SF2 SI4 SF4 SI7 SF7 SI6 SF6 SI8 SF8 ✓

Question **3**
Complete
Marked out of 1.00
Flag question

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:

R1	R2	R3	R4
2	1	2	0

Sarcina	Alocare curentp				Necesar Maxim				Necesar pentru finalizare			
	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

Este acest sistem blocat în prezent sau poate vreun proces să devină blocat?

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

Dacă nu ar fi blocat, specificați o secvență de execuție posibilă.

Dacă sosește o cerere din partea sarcinii S1 pentru resursele (0, 4, 2, 0), cererea poate fi imediat

acceptată?

Justificați dacă da sau de ce nu?

Dacă da, stabiliți secvența de execuție.

Dacă sosește o cerere dintr-un proces S2 pentru resursele (0, 1, 2, 0), ar trebui ca cererea să fie imediat acceptată?

Justificați dacă da sau de ce nu?

Dacă da, stabiliți secvența de execuție.

Aplicam algoritmul pentru detectarea blocării și observăm că pentru sarcina S2, pentru resursa R2 se alocă 0, necesar pentru finalizare fiind 7, așa că procesul devine blocat.

Question **4**

Complete

Marked out of
1.00

🚩 Flag question

Descrieți comparativ algoritmul optim de partiție a unui microprogram descries prin graf de dependențe de date vs algoritmul euristic de partiție.

Pentru algoritmul optim de partiție minima are o operație MP-completa și se bazează pe generarea setului de microoperații a tuturor IC posibile, iar pentru fiecare dintre ele se analizează influența asupra generării setului de microoperații disponibile următor, iar în cazul partiției euristice, algoritmul este mai rapid (înșă partitionarea nu e optimă) și se bazează pe ordonarea microoperațiilor în funcție de succesorii din graful de dependențe de date.

Question **5**

Complete

Mark 1.00 out of
1.00

🚩 Flag question

Descrieți principiul de funcționare a sistemelor de tip Data Flow . Comentați componentele principale a le arhitecturii.

Modelul data flow (se bazează pe fluxul de date), bazându-se pe faptul că execuția unei instrucțiuni se face numai după ce toți operandii devin disponibili - se va realiza un graf direcționat de conexiune între operatori și operanți, fiecărei variabile fiindu-i asignat un token de disponibilitate. Abia atunci când toți tokenii sunt disponibili pentru un nod, instrucțiunea se declanșează și își începe execuția.

Sistemul nu dispune de un program counter.

Timpul total de execuție este dat de suma de timpi de propagare prin elementele de execuție.

Componente importante ale arhitecturii sunt coada de token-uri și unitatea care creează tokens. Dataflow-ul poate fi de mai multe tipuri:

Abordare statică:

- trebuie să se trimită confirmare că un element de procesare a primit date
- orice execuție a unei bucle se face ca și cum am avea iterații diferite
- se activează cele care au token pe intrare și nu a apărut token pe ieșire

Abordare dinamică:

- putem să avem subgrafuri reentrante
- informațiile de context definesc nodul

Interpretare tip U: token = nume activitate(cuprinde tag) + date. Câmpul contextual poate fi el însuși nume de activitate.

Comment:

Question 6

Correct

Mark 1.00 out of 1.00

Flag question

Fie un procesor care implementează o structura paralela pipeline de citire interpretare executie pentru procesare suprascalară , cu patru unitati paralele.

Presupunând un ciclu de instrucțiuni contine

-citire care necesita o singura perioada de ceas

-decodificarea instrucțiunii necesita două perioade de ceas

-executia instrucțiunii necesita doua perioade de ceas

Avem o secvență de 33000000 de instrucțiuni.

Considerand freventa ceasului in Ghz de 3.5.

Calculati durata de executie a secventei de program in milisecunde

Answer: ✖

The correct answer is: 4.71

Comment:

răspuns corect a avut toleranța 0.01 de asta era greșit

Question 7

Incorrect

Mark 0.00 out of 1.00

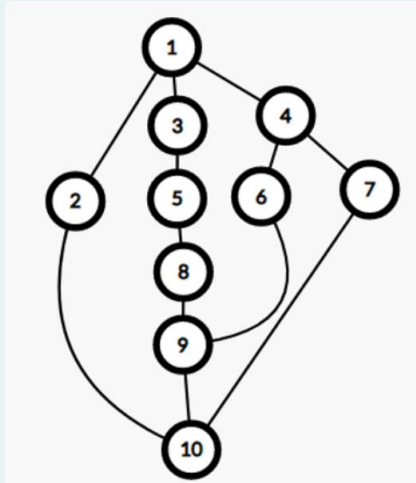
Flag question

Un sistem multiprocesor dispune de 240 procesoare.

Avem un JOB impartit in 10 sarcini.

Durata de executie secventiala a unei sarcini este de 386.

Dependenta de date intre sarcini este data in figura alaturata.



Considerand ca sarcinile au o zona de secventialitate in procent de 5%.

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

Answer: ✖

Question **8**

Complete

Marked out of
1.00

🚩 Flag question

Explicați în cuvinte - pseudocod - algoritmul de excludere mutuală bazat pe XCHG

Continutul registrului r cu celula de memorie m este interschimbat prin de instrucțiunea xchg, ca o operație atomică.

start sarcina

repetă

 xchg(r, m)

pană când ri

secțiune critică

xchg(ri, m)

restul sarcinii, unde ri = variabilă locală, m= variabilă globală (variabilele pot lua valorile 0, 1)

Atunci când o sarcină intră în secțiunea critică, ri devine 1.

Pentru excluderea mutuală, trebuie verificată relația: $m + \text{suma}(ri) = 1$

[Dashboard](#) / [My courses](#) / [03-ACS-L-A4-S1-APPa-C1](#) / [General](#) / [Examen Final](#)**Started on** Monday, 8 February 2021, 10:15 AM**State** Finished**Completed on** Monday, 8 February 2021, 11:30 AM**Time taken** 1 hour 15 mins**Grade** Not yet gradedQuestion **1**

Complete

Mark 0.90 out of 1.00

Descrieți principiul de funcționare a sistemelor de tip Data Flow . Comentați componentele principale a le arhitecturii.

Sistemele de tip Data Flow au un model ce este bazat pe fluxul de date, in cadrul caruia executia unei instructiuni de prelucrare depinde de disponibilitatea operandului. In cadrul acestor sisteme nu avem Program Counter. Specifica efectuarea unor operatii imediat ce vom avea operanzii disponibili si din aceasta cauza nu avem nevoie de contorul de program. De asemenea, nu este nevoie de adrese pentru memorarea variabilelor iar operatia se efectueaza numai atunci cand token-ul este prezent pe ambele intrari ale operandului. Datele se stocheaza in cadrul instructiunilor, adica vom mentine un token cu valoare binara in cadrul instructiunii pentru fiecare data. O unitate specializata va verifica disponibilitatea datelor. Ca avantaje avem faptul ca avem un paralelism ridicat si o viteza de executie mai mare. De asemenea este usor de implementat partea de comunicatie si sincronizare, insa vom avea overhead mare pentru control, iar manipularea structurilor de date va fi dificila. Avem $\text{Token} = \{\text{Data}, \text{Tag Destinatie}, \text{marker}\}$ si $\text{Match} = \{\text{Tag}, \text{Destinatie}\}$. Avem Df_1, Df_2, \dots, Df_n ce reprezinta retea de masini de baza ce sunt interconectate printr-o retea de comutare; Token Queue cu care pe masura ce vom avea token-uri, ele vor fi adaugate in coada Matching Unit - elementul de prelucrare are un token pe fiecare ramura; Instruction store - ia elementele de prelucrare si le pune pe o unitate de procesare

Comment:

Question **2**

Complete

Marked out of 1.00

Explicați în cuvinte - pseudocod - algoritmul de excludere mutuală bazat pe XCHG

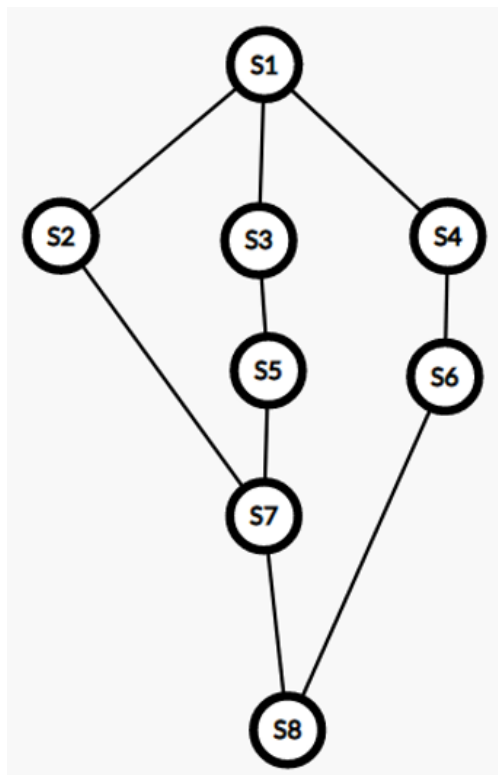
Pentru început, vom spune faptul că instrucțiunea $XCHG(r, m)$ interschimbă fix ca o operație atomică, conținutul registrului r cu celula de memorie m . R_i va fi o variabilă locală sarcinii S_i și va fi inițializată cu 0, iar m va fi o variabilă globală, ce va fi comună tuturor sarcinilor și va fi inițializată cu 1. Valorile celor două pot fi 0 și 1. Atunci când un proces găsește $m = 1$ și intră în secțiunea critică și m va avea valoarea 0. Atunci când iese din secțiunea critică, m va deveni din nou 1, pentru a permite și altor sarcini să intre în secțiunea critică. M este practic un semafor care dă voie unei singure sarcini să parcurgă secțiunea critică la un moment de timp. Când o sarcină va intra în secțiunea critică $r_i = 1$. Practic, la orice moment de timp pentru a avea excludere mutuală, în orice moment $m + \text{Suma sarcinilor}$ va fi egală cu 1.

Question 3

Correct

Mark 1.00 out of 1.00

Fiind dat sistemul de sarcini



Care secvență de execuție este validă ?

- ☐ a. S1 SF1 S2 SI3 SF3 SI5 SI7 SF2 SI4 SF4 SF5 SF7 SI6 SF6 SI8 SF8
- ☒ b. S1 SF1 S2 SF2 SI3 SF3 SI4 SF4 SI5 SF5 SI6 SF6 SI7 SF7 SI8 SF8 ✓
- ☐ c. S1 SF1 S2 SI3 SF3 SI4 SF4 SI5 SI7 SF2 SF5 SI6 SF6 SF7 SI8 SF8
- ☐ d. S1 SF1 S2 SI3 SF3 SI5 SI7 SF2 SF5 SI4 SF4 SI6 SF7 SF6 SI8 SF8

Your answer is correct.

The correct answer is:

S1 SF1 S2 SF2 SI3 SF3 SI4 SF4 SI5 SF5 SI6 SF6 SI7 SF7 SI8 SF8

Question **4**

Complete

Marked out of 1.00

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:

R1	R2	R3	R4
2	1	2	0

	Alocare curentp				Necesar Maxim				Necesar pentru finalizare			
Sarcina	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

Este acest sistem blocat în prezent sau poate vreun proces să devină blocat?

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

Dacă nu ar fi blocat, specificați o secvență de execuție posibilă.

Dacă sosește o cerere din partea sarcinii S1 pentru resursele (0, 4, 2, 0), cererea poate fi imediat acceptată?

Justificați dacă da sau de ce nu?

Dacă da, stabiliți secvența de execuție.

Dacă sosește o cerere dintr-un proces S2 pentru resursele (0, 1, 2, 0), ar trebui ca cererea să fie imediat acceptată?

Justificați dacă da sau de ce nu?

Dacă da, stabiliți secvența de execuție.

a) Sistemul este blocat în prezent, deoarece S2 pentru R2 va da un 0, însa este nevoie de 7 pentru finalizare

b) Dacă din partea sarcinii S1 pentru (0,4,2,0) cererea nu poate fi imediat acceptata, deoarece resursele pe care S1 incearca sa le rezerve sunt ocupate.

c) Dacă soseste o cerere dintr-un proces S2 pentru (0,1,2,0) cererea ar putea fi acceptata, pentru ca se incadreaza in resursele disponibile in sistemul descris in linia 2.

S1-> S2 -> S3 -> S4 -> S5

Question **5**

Complete

Marked out of 1.00

Descrieți comparativ algoritmul optim de partiție a unui microprogram descris prin graf de dependențe de date vs algoritmul euristic de partiție.

În cadrul algoritmului optim de partiție a unui microprogram, descris prin graf de dependente de date, se va realiza partiția optimă prin generarea din setul de microoperații uODk a tuturor uIC posibile, pentru fiecare dintre ele se va analiza influența pe care o are asupra generării setului de microoperații disponibile următor, uODk + 1. În cazul algoritmului euristic de partiție, el este bazat pe ordonarea a microoperațiilor din setul disponibil uODk în funcție de succesorii în graful de dependente de date. În cadrul algoritmului euristic nu se va genera însă o partiție optimă, dar va fi mult mai rapidă în unele situații.

partiție euristica = bazată pe ordonarea operațiilor în funcție de succesorii → complexitate liniară

algoritm optim de partiție = MP completă

Question 6

Incorrect

Mark 0.00 out of 1.00

Considerând că în urma analizei microoperațiilor dintr-un subloc au rezultat microinstrucțiunile complete, mIC1,...mIC5 care conțin microoperațiile ca în tabelul alăturat:

mIC1 = mo1 mo2 mo3 mo4 mo5 mo6

mIC2 = mo3 mo7 mo8 mo9

mIC3 = mo1 mo2 mo8 mo9 mo10

mIC4 = mo4 mo8 mo11

mIC5 = mo6 mo8

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

- ☐ a. Câmp 1: (mo1)
Câmp 2: (mo2)
Câmp 3: (mo3)
Câmp 4: (mo4 mo7 mo9 mo10 mo11)
Câmp 5: (mo5)
Câmp 6: (mo6)
Câmp 7: (mo8)
- ☐ b. Câmp 1: (mo1)
Câmp 2: (mo2)
Câmp 3: (mo3)
Câmp 4: (mo4 mo9 mo10)
Câmp 5: (mo5 mo11)
Câmp 6: (mo6 mo7)
Câmp 7: (mo8)
- ☒ c. Câmp 1: (mo1)
Câmp 2: (mo2)
Câmp 3: (mo3)
Câmp 4: (mo4 mo9)
Câmp 5: (mo5 mo11)
Câmp 6: (mo6 mo7 mo10)
Câmp 7: (mo8)
- ☐ d. Câmp 1: (mo1)
Câmp 2: (mo2)
Câmp 3: (mo3)
Câmp 4: (mo4)
Câmp 5: (mo5 mo7 mo10)
Câmp 6: (mo6 mo9 mo11)
Câmp 7: (mo8)



Your answer is incorrect.

The correct answer is:

Câmp 1: (mo1)

Câmp 2: (mo2)

Câmp 3: (mo3)

Câmp 4: (mo4)

Câmp 5: (mo5 mo7 mo10)

Câmp 6: (mo6 mo9 mo11)

Câmp 7: (mo8)

Question **7**

Correct

Mark 1.00 out of 1.00

Fie un procesor care implementează o structura paralela pipeline de citire interpretare executie pentru procesare suprascalară , cu patru unitati paralele.

Presupunând un ciclu de instrucțiuni contine

-citire care necesita o singura perioada de ceas

-decodificarea instructiunii necesita două perioade de ceas

-executia instructiunii necesita doua perioade de ceas

Avem o secvență de 3000000 de instrucțiuni.

Considerand freventa ceasului in Ghz de 1.

Calculati durata de executie a secventei de program in milisecunde

Answer: ✖

The correct answer is: 1.50

Comment:

- răspuns corect a luat virgula ca separator de mii și vedea răspunsul 1500.005

Question 8

Incorrect

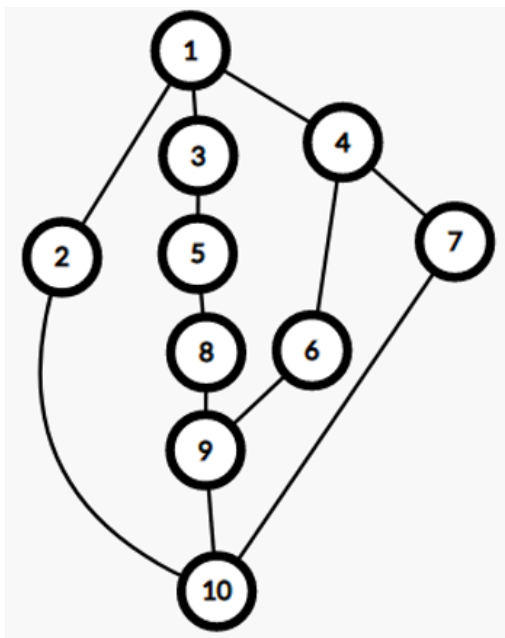
Mark 0.00 out of 1.00

Un sistem multiprocesor dispune de 600 procesoare.

Avem un JOB impartit in 10 sarcini.

Durata de executie secventiala a unei sarcini este de 29.

Dependenta de date intre sarcini este data in figura alaturata.



Considerand ca sarcinile au o zona de secventialitate in procent de 2%.

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

Answer: ❌

The correct answer is: 73.35

◀ [Prezentari curs APP](#)

Jump to...

[OpenMP Overview](#) ▶