



Блокади



Оперативни системи 2014
Аудиторски вежби

Блокада (Deadlock, Ќорсокак)

- ▶ Дефиниција: Во мултипрограмска околина може да дојде до ситуација кога повеќе процеси, кои чекаат на ресурси за да продолжат, никогаш не ги добијат тие ресурси бидејќи нив ги „држат“ други процеси кои исто така чекаат. Ваквата ситуација се нарекува **блокада** или **ќорсокак**.
- ▶ Иако постојат методи за откривање и разрешување на вакви ситуации, повеќето ОС немаат механизми за спречување на ќорсокак.

Модел на компјутерски систем (КС)

- ▶ **Компјутерски систем (КС)** може да се моделира како конечен број на **ресурси** и **процеси** кои се натпреваруваат за контрола над тие ресурси.
- ▶ Ресурсите можат да бидат поделени во класи составени од идентични ресурси (на пример: 2 CPUs, 5 принтери, итн.).
- ▶ Можат да бидат физички (дискови, принтери) или логички (семафори, датотеки и монитори).

Ресурси

- ▶ **Ресурс** може да биде некој хардверски уред (принтер, скенер, итн.) или пак некое парче информација (на пример: заклучен запис во база на податоци или споделени податоци во меморија).
- ▶ **Процесите** се оние кои ги користат ресурсите. Но, пред да започне да го користи, процесот првин мора да го побара ресурсот. По употребата, мора да го ослободи.
- ▶ Оттука, низата чекори на користење ресурс од страна на процес се:
 - ▶ барање
 - ▶ употреба
 - ▶ ослободување
- ▶ Барање и ослободување ресурс се системски повици.

Граф за алокација на ресурси

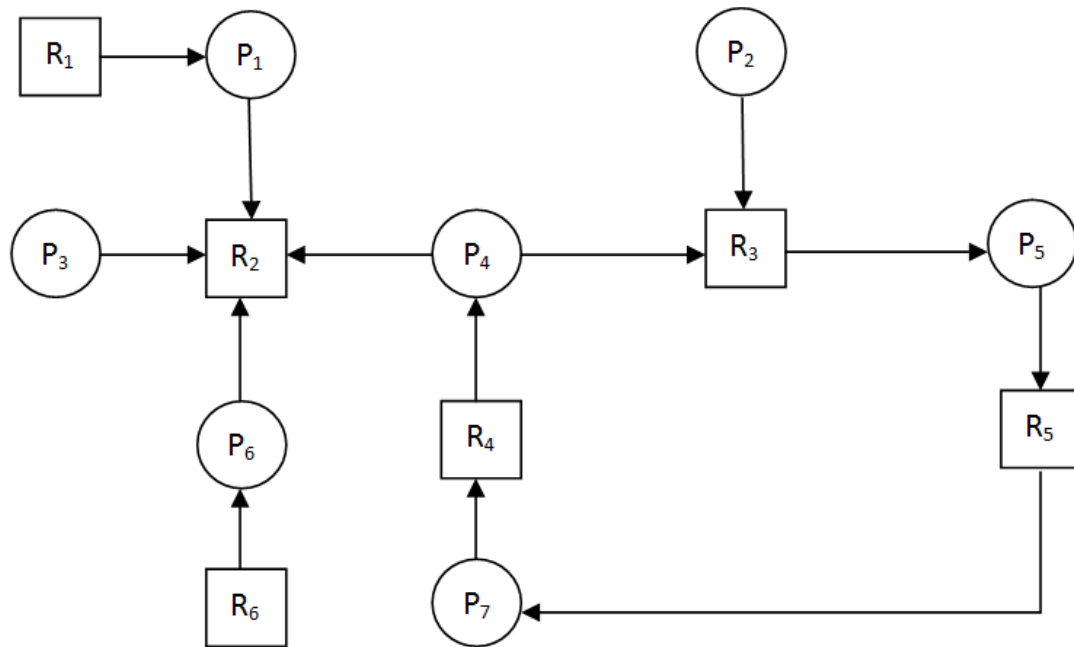
- ▶ Графот за алокација на ресурси е еден од начините за откривање ќорсокак во системите.
 - ▶ Претставува насочен граф
- ▶ Ако графот нема циклуси, тогаш велиме дека системот е слободен од ќорсокак.
- ▶ Доколку графот за алокација на ресурси има циклуси, тогаш велиме дека е можно системот да има ќорсокак (потребен услов).
- ▶ Ако секој ресурс има само по една инстанца, тогаш тоа е и доволен услов и со сигурност велиме дека системот има ќорсокак.

Задача 1

- ▶ Нека имаме седум процеси P_1, P_2, \dots, P_7 и шест ресурси R_1, R_2, \dots, R_6 .
- ▶ Состојбата за тоа кои ресурси се зафатени од кои процеси и кои се побарани да бидат зафатени е дадена со следново:
 1. Процесот P_1 го држи ресурсот R_1 и го побарува R_2 ,
 2. Процесот P_2 не држи ниту еден ресурс, само го побарува R_3 ,
 3. Исто и процесот P_3 само го побарува ресурсот R_2 ,
 4. Процесот P_4 го држи ресурсот R_4 и ги побарува R_2 и R_3
 5. Процесот P_5 го држи ресурсот R_3 и го побарува R_5 ,
 6. Процесот P_6 го држи ресурсот R_6 и го побарува R_2 ,
 7. Процесот P_7 го држи ресурсот R_5 и го побарува R_4
- ▶ Да се провери дали овој систем е безбеден од ќорсокак.

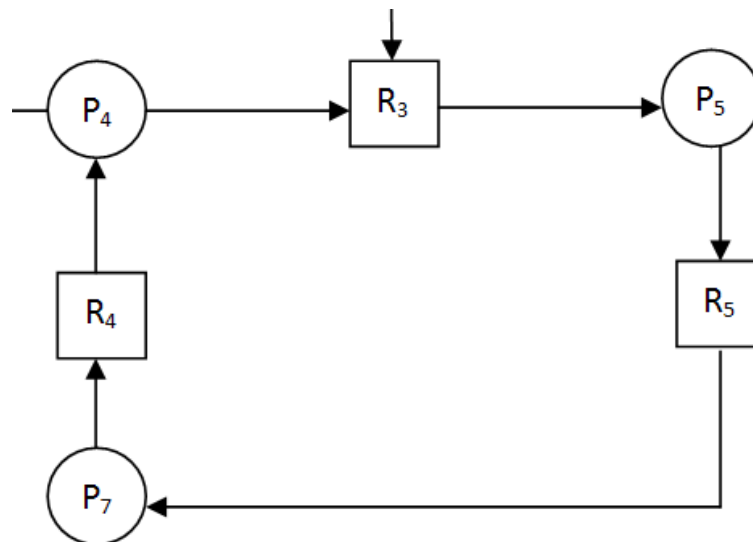
Решение:

- ▶ За да провериме дали дадениот систем е сигурен од ќорсокак, ќе конструираме граф за алокација на ресурси.



Решение:

- Поради тоа што претставениот граф содржи циклус, следува дека нашиот систем има ќорсокак и тој е претставен со процесите и ресурсите дадени на сликата.



Алгоритам за детекција на блокади

- ▶ Не секогаш имаме едноставни системи, каде ресурсите имаат само по една инстанца и кои може едноставно да ги претставиме со граф, со цел да видиме дали постои циклус или не.
- ▶ Многу често во системите каде ресурсите имаат повеќе од една инстанца за детекција на блокада се користат алгоритми кои го прават тоа.

Алгоритам за детекција на ќорсокак во системи каде ресурсите имаат повеќе инстанци

- ▶ Овој алгоритам за откривање на ќорсокак е базиран на **матрици**, каде во **редиците** се претставени **процесите**, а во **колоните** ресурсите.
 - ▶ **E**: вектор со бројот на сите постоечки инстанци од класите на ресурси во системот - [5 3 1]
 - ▶ **A**: вектор на слободни инстанци од класите на ресурсите во системот - [2 0 1]
 - ▶ **C**: матрица на моментално алоцираните ресурси од страна на процесите
 - ▶ **R**: матрица на побарувани ресурси од страна на процесите

Алгоритам за детекција на ќорсокак во системи каде ресурсите имаат повеќе инстанци

- ▶ Секој процес во системот на почеток на алгоритамот велиме дека иницијално е неозначен (**не е завршен**).
- ▶ Како што алгоритамот е во прогрес, така и процесите еден по еден ќе се означуваат и доколку заврши алгоритамот и сите процеси во системот се означени (**завршени**), велиме дека системот нема ќорсокак.
- ▶ Доколку пак алгоритамот заврши, сите процеси кои ќе останат неозначени велиме дека се **блокирани** и **системот има ќорсокак**.

Алгоритам за детекција на ќорсокак во системи каде ресурсите имаат повеќе инстанци

▶ Алгоритамот е следен:

- ▶ Најди неозначен процес P_i , за кој i -та редица од матрицата R е помала или еднаква на A .
 - ▶ Произволен процес кој има потреба од помалку ресурси од тековно слободните
- ▶ Ако е најден таков процес, додади ја i -та редица од матрицата C на векторот A и врати се на чекор 1.
 - ▶ Процесот ги добил ресурсите, завршил со својата работа и потоа ги ослободува сите алоцирани ресурси
 - оние кои штотуку ги добил (i -тата редица од R) + оние кои претходно му биле алоцирани (i -тата редица од C)
- ▶ Ако не постои таков процес, алгоритамот завршува.

Задача 2

- ▶ Нека се дадени три процеси и четири класи ресурси, кои што се именувани како: принтери, плотери, скенери и ДВД-читачи.
- ▶ Системот има на располагање 4, 2, 3, 1 инстанци од соодветните четири класи ресурси.
- ▶ Првиот процес има само еден скенер, вториот процес има два принтери и ДВД-читач и третиот процес има плотер и два скенери.
- ▶ Секој од процесите има потреба од дополнителни ресурси како што е тоа прикажано со матрицата R.
- ▶ Со помош на алгоритамот за детекција на блокада да се утврди дали постои блокада во овој систем.

$$R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

Решение:

- ▶ Векторот на расположливи инстанци од секој од ресурсите е даден со E :

$$E = [4 \quad 2 \quad 3 \quad 1]$$

- ▶ Матрицата на зафатени инстанци за секој од процесите е дадена со C :

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix}$$

Решение:

- ▶ Матрицата на потребни инстанци за секој од процесите да ја заврши својата работа е дадена со R

$$R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

- ▶ Векторот на слободни инстанци за секој од ресурсите е даден со A

$$A = [2 \quad 1 \quad 0 \quad 0]$$

Решение:

- ▶ За да започнеме со алгоритмот за детекција на блокади, треба да најдеме процес чие што побарување на инстанци од соодветните ресурси ќе биде задоволено.
- ▶ Првиот процес: не може да биде задоволен поради тоа што нема слободен ДВД-читач за него.
- ▶ Вториот процес: не може да биде задоволен поради тоа што нема слободен скенер за него.
- ▶ Третиот процес: има доволно слободни инстанци од ресурсите кои ги бара.

Решение:

- ▶ Значи, прв започнува да се извршува третиот процес, кој откако ќе заврши ќе ги ослободи сите инстанци од ресурсите кои ги имаше зафатено (0 1 2 0).
- ▶ Според тоа векторот на слободни инстанци сега ќе се промени и тој ќе биде:

$$A = [2 \quad 2 \quad 2 \quad 0]$$

Решение:

- ▶ Следен ќе може да се извршува процесот 2 и откако ќе заврши тој ќе ги врати инстанците од ресурсите кои ги имаше претходно зафатено (2 0 0 1).
- ▶ Сега векторот A ќе биде:

$$A = [4 \quad 2 \quad 2 \quad 1]$$

- ▶ И на крај и за процесот 1 има доволно инстанци од ресурсите кои ги побарува за да се изврши и системот да заврши **без да се случи блокада**.

Задача 2

- ▶ Што ќе се случеше доколку во претходниот систем, побарувањата на процесот 2 беа 1 инстанца од ДВД-читачот, 2 инстанции од принтерот и 1 инстанца од плотерот?

Решение:

- ▶ Матрицата R би изгледала:

$$R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

- ▶ Векторот A е:

$$A = [2 \quad 1 \quad 0 \quad 0]$$

- ▶ Откако ќе заврши третиот процес, ниту еден од преостанатите процесите нема да може да започне да се извршува, затоа што нема доволно слободни ресурси кои се потребни за процесите да завршат.
- ▶ Поради тоа, **системот ќе влезе во ќорсокак.**

Банкаров алгоритам

- ▶ Нека n е број на процеси, m е број на ресурси.
- ▶ Алгоритамот користи низа од податочни структури:
 - ▶ вектор $Available\{m\}$ со должина m
 - ▶ $Available[j]=k$, ако k единици од ресурсот R_j се слободни;
 - ▶ матрица $Max\{n \times m\}$
 - ▶ $Max[i,j]=k$, ако процесот P_i бара најмногу k единици од ресурсот R_j
 - ▶ матрица на алокација $Allocation\{n \times m\}$
 - ▶ $Allocation[i,j]=k$, ако на процесот P_i му се моментално доделени k единици од ресурсот R_j ;
 - ▶ матрица $Need\{n \times m\}$
 - ▶ $Need[i,j]=k$, ако P_i побарува уште k единици од ресурсот R_j ;
- ▶ Притоа важи:
 - ▶ $Need[i,j] = Max[i,j] - Allocation[i,j]$;

Банкаровиот алгоритам е следниот:

- ▶ Слободни ресурси во итерацијата $iter$: $Work_{iter}$
- ▶ 1. Иницијално:
 - ▶ $iter := 0$;
 - ▶ $Work_0 := Available$;
 - ▶ $Finish[i] := false$; (за $i = 1, 2, \dots, n$).
- ▶ 2. Најди i за кое:
 - ▶ $Finish[i] = false$;
 - ▶ $Need[i] \leq Work_{iter}$;
 - ▶ Ако не постои такво i , оди на чекор 4.
- ▶ 3. $Work_{iter+1} := Work_{iter} + Allocation[i, *]$;
 - ▶ $Finish[i] := true$;
 - ▶ $iter++$;
 - ▶ Врати се на чекор 2.
- ▶ 4. Ако $Finish[i] = true$; за секое i , тогаш системот е безбеден од ќорсокак.

Задача 3

- ▶ Нека е даден систем со 12 инстанци (единици) од еден ист ресурс и 3 процеси А, В и С. Која од следниве состојби е сигурна од ќорсокак?

а)

	Max	Current
A	10	5
B	4	2
C	9	3

б)

	Max	Current
A	10	5
B	4	2
C	9	2

Решение:

	Max	Current
A	10	5
B	4	2
C	9	3

- ▶ а) Состојбата не е сигурна. Во овој момент само на процесот В може да му се доделат 2-те слободни инстанци од ресурсот, но по негово завршување системот влегува во ќорсокак.

Решение:

	Max	Current
A	10	5
B	4	2
C	9	2

- ▶ б) Состојбата е сигурна. Во овој момент само на процесот В може да му се доделат 2 од 3-те слободни инстанци од ресурсот. По негово завршување, слободните инстанци (вкупно 5) се доделуваат на процесот А и на крај на процесот С. Оттука, системот не влегува во ќорсокак, т.е. е безбеден.

Задача 4

- ▶ Нека е даден систем со 5 процеси: P_0 до P_4 , 3 типа на ресурси А, В и С кои имаат по 10, 5 и 7 единици соодветно.
- ▶ Определете дали системот се наоѓа во состојба безбедна од ќорсокак, ако во даден момент системот ја има следната состојба:

Задача 4

	Allocation			Max			Available			Finish
	A	B	C	A	B	C	A	B	C	
P0	0	1	0	7	5	3	3	3	2	0
P1	2	0	0	3	2	2				0
P2	3	0	2	9	0	2				0
P3	2	1	1	2	2	2				0
P4	0	0	2	4	3	3				0

Решение:

- ▶ Матрицата Need=Max-Allocation за оваа состојба на системот е:

	Need		
	A	B	C
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

Решение:

- ▶ Бидејќи $Need_1 \leq Available$, процесот P_1 може да се изврши.
- ▶ Откако P_1 ќе ги ослободи ресурсите, новата состојба ќе биде:

	Allocation			Max			Work			Finish
	A	B	C	A	B	C	A	B	C	
P0	0	1	0	7	5	3	5	3	2	0
P1	0	0	0	3	2	2				1
P2	3	0	2	9	0	2				0
P3	2	1	1	2	2	2				0
P4	0	0	2	4	3	3				0

Решение:

- ▶ Следно, важи $Need_3 \leq Work$, па може да се изврши P_3 , итн.
- ▶ Процесите можат да се извршат во следниот редослед: P_1, P_3, P_4, P_2, P_0 , при што ќе бидат задоволени побарувањата на процесите.
- ▶ Значи, состојбата е сигурна од ќорсокак.

Задача 5

- ▶ Нека еден систем има 4 инстанци од ист ресурс и 3 процеси. Секој процес може да побарува најмногу две инстанци.
- ▶ Покажи дека системот е безбеден од ќорсокак.

Решение:

- ▶ Да претпоставиме дека системот е во ќорсокак.
- ▶ Тоа значи дека секој процес има по еден ресурс и чека на уште еден.
- ▶ Бидејќи има вкупно 4 ресурси, има уште еден слободен, па може да се изврши процес кој потоа ќе ослободи ресурс.
- ▶ Значи, не може да дојде до ќорсокак.

Задача 6

- ▶ Нека банкаровиот алгоритам се извршува кај систем со m типови на ресурси и n процеси.
- ▶ Бројот на операции потребни да се извршат за да се провери дали системот е безбеден од ќорсокак е од ред $m^a \times n^b$.
- ▶ Кои се вредностите на a и b ?

Решение:

- ▶ Чекор 2 се извршува $n+(n-1)+(n-2)+\dots+3+2+1=n(n+1)/2$ пати.
- ▶ Во секое извршување на чекор 2 се испитуваат по m вредности (за секој ресурс).
- ▶ Значи, има вкупно $m \times n(n+1)/2$ операции, т.е. $a=1$ и $b=2$.