

# OPERACIJSKI SISTEMI, 28.8.2015

## 1. Naštejte glavne štiri funkcionalnosti (stebre) vsakega OS-a.

1. Procesi in niti
2. Upravljanje s pomnilnikom
3. Razporejanje procesov
4. Komunikacija z V/I napravami

### V katero izmed njih spada Bančniški algoritem?

Razporejanje procesov.

## 2. Opišite, kako deluje algoritem za zaznavo smrtne objema. Ne pozabite na začetku opisati, kakšen je vhod v algoritem, na koncu pa kakšen je izhod.

1. Vhodi v algoritem so:
  - Matrika A (allocation matrix)
  - Matrika C (claim matrix)
  - Vektor R (resource vector)
2. Seštejemo vrstice v stolpcih A in jih odštejemo od vrstice R, da dobimo vektor V (available vector)
3. Odštejemo matriko A od matrike C, da dobimo C-A matriko
4. Primerjamo V z vrsticam C-A, dokler ne najdemo vrstice C-A  $\leq$  V
5. V - vrstica v C-A (tista, ki ustreza pogojem iz 4.)
6. Prištejemo vrstico z enakim indexom iz C v V
7. Ponavljamo korake 4., 5., 6. dokler ne pride do tega, da je V večji od R ali pa da hkrati dve vrstici prideta v V (dve vrstici v C-A naenkrat sta  $\leq$  V) - pogoj za deadlock
8. Izhod iz algoritma je *true* ali *false* (program se lahko izvede ali pa ne)

## 3. Sveženjsko multiprogramiranje in delitev procesorskega časa uporabljata multiprogramiranje, a z različnim ciljem. Kaj je glavni cilj sveženjskega multiprogramiranja in kaj delitve procesorskega časa?

Glavni cilj sveženjskega multiprogramiranja je maksimizirati uporabo procesorja, glavni cilj delitve procesorskega časa pa je minimiziranje odzivnega časa.

## 4. OS ima štiri nadzorne podatkovne strukture, ki hranijo informacije o trenutnem stanju vsakega procesa in vira. Katere so te štiri nadzorne strukture?

Pomnilnik, V/I, datoteka in proces.

**5. Katere tri mehanizme smo spoznali, ki zagotavljajo podporo vzajemnemu izključevanju znotraj samega OS-a?**

Semaforji, monitorji in predajanje sporočil.

**Katerega od teh mahanizmov smo uporabljali na vajah?**

Semafor (lockfile)

**6. Koncept procesa smo nadgradili s konceptom niti. Katera stanja nitnega modela stanj so še vedno upravljana na nivoju procesa?**

- Izhod
- Blokiran ustavljen (začasno odstranjen)
- Pripravljen ustavljen

**Kaj pravi Amdahlov zakon?**

Pospešitev je enaka času izvedbe programa na enem procesorju deljeno s časom izvedbe programa na N procesorjih.

**7. Pravilnost delovanja realno-časovnega OS-a je odvisna od dveh stvari. katerih?**

Odvisna je od logičnega rezultata procesiranja in časa v katerem je rezultat na voljo.

**Katere štiri pristope k razporejanju v realnem času smo spoznali?**

- Staticno tabelarno gnano
- Staticno prioriteto gnano preklopno
- Dinamično plansko osnovano
- Dinamično celostno

8. Pri upravljanju pomožnega pomnilnika je zelo pomemben postopek zaseganja prostora za datoteke. Katere tri metode zaseganja smo spoznali?

- 1. Zvezno zaseganje
- 2. Verižno zaseganje
- 3. Indeksirano zaseganje

Kako izgledajo tabele FAT pri vsaki izmed njih?

- 1. Zvezno zaseganje

File Allocation Table

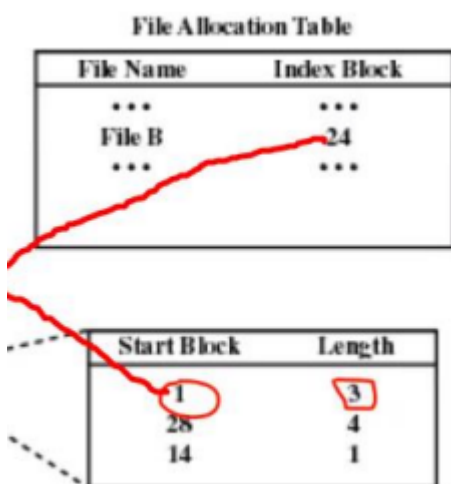
File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

- 2. Verižno zaseganje

File Allocation Table

File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...

- 3. Indeksirano zaseganje



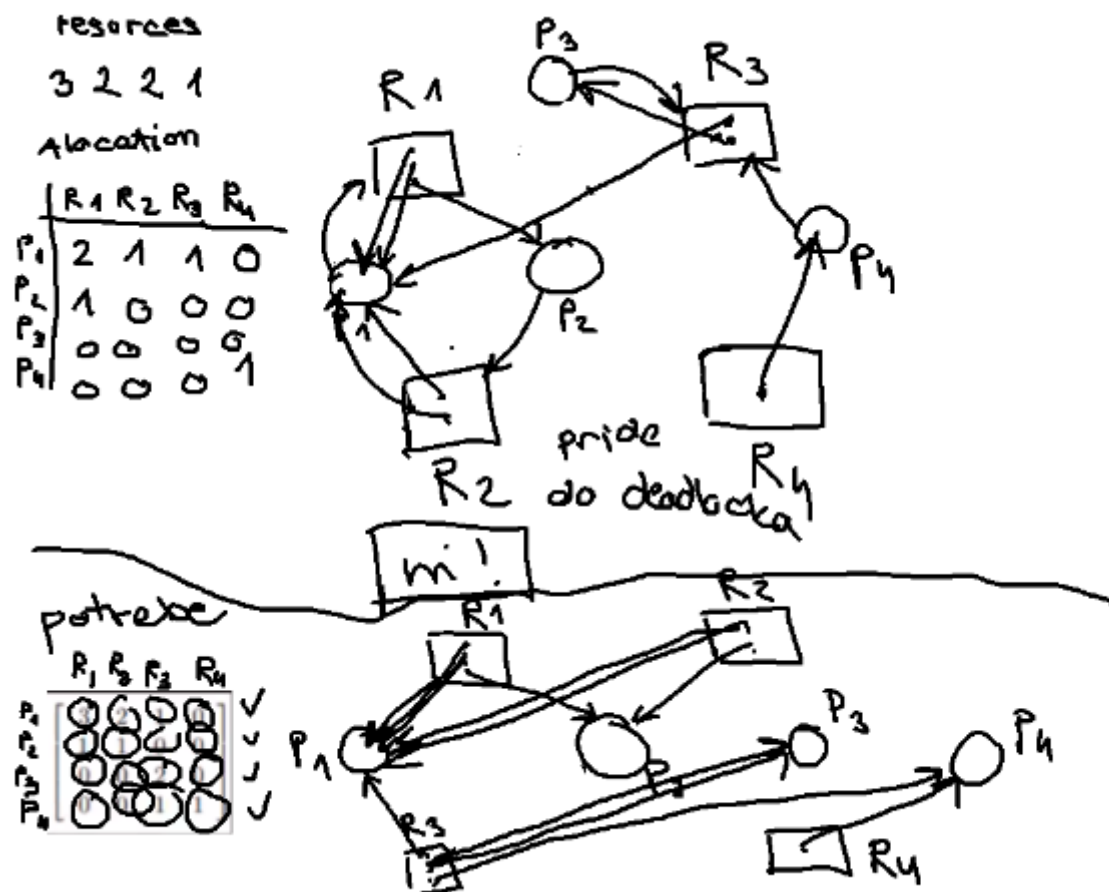
9. V času načrtovanja sistema predvidimo, da bodo na računalniku tekli štiri procesi (P1-P4). Ti bodo med izvajanjem potrebovali različne vire (R1-R4). Števnost virov v računalniškem sistemu je: R1 – 3, R2 – 2, R3 – 2, R4 – 1. Potrebe so podane v naslednji matriki (v vrsticah so procesi, v stolpcih pa viri):

$$\begin{bmatrix} 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Trenutno ima P1 v lasti dva vira R1, po en vir R2 in R3, P2 ima v lasti vir R1, P4 pa ima v lasti R4.

S pomočjo grafa alociranja virov preverite, ali je prišlo do smrtne objema? Narišite graf in odgovor argumentirajte.

Pride do deadlocka





11. Imamo en procesor in več procesov, ki jih želimo izvajati na procesorju. Čas prispetja in čas izvajanja posameznega procesa je sledeč:

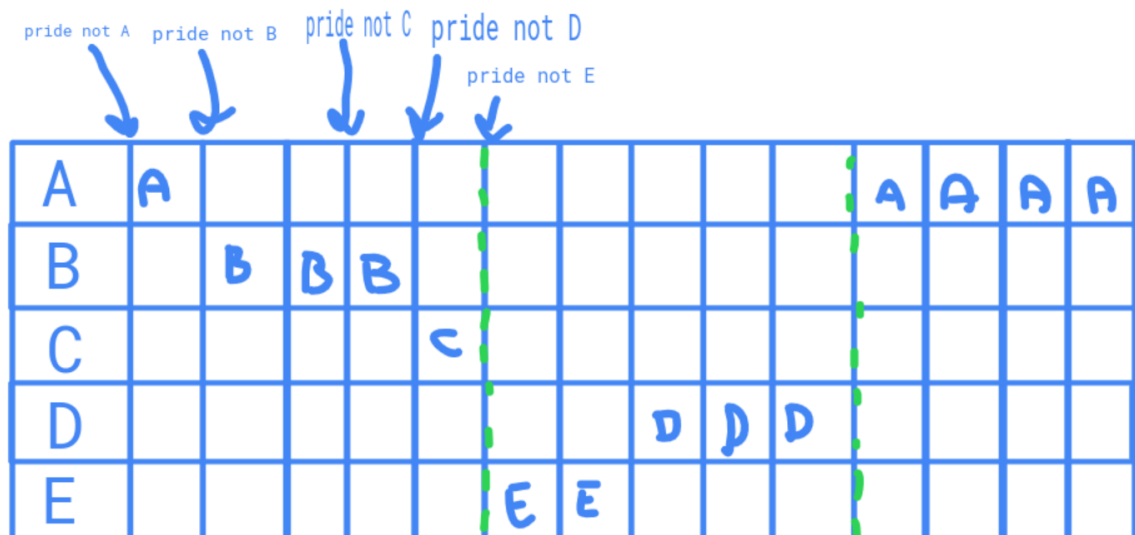
proces	čas prispetja	čas izvajanja
A	0	5
B	1	3
C	3	1
D	4	3
E	5	2

Skicirajte kratkoročno razporejanje:

- po kriteriju najprej najkrajši proces (angl. Shortest Process Next)



- po kriteriju najprej tisti z najkrajšim preostankom časa (angl. Shortest Remaining Time)



- Primerjajte rezultate obeh algoritmov – izračunajte in primerjajte povprečen normaliziran obračalni čas. Kateri algoritem je bil v danem primeru boljši? (od prihoda do konca izvedbe / pričakovan čas izvajanja) za vsak proces in potem povprečje vseh rezultatov

$$\text{Shortest Process Next} \rightarrow (5/5 + 10/3 + 3/1 + 10/3 + 3/2) / 5 =$$



### 13. BASH

- **Kako bi s pomočjo preusmerjenja zapisali v datoteko test.txt besedilo test?**

```
bash echo test >test.txt
```

- **Ustvarite trdo povezavo z imenom test2.txt na datoteko test.txt**

```
bash ln test.txt test2.txt
```

- **Kako bi preverili koliko trdih povezav ima datoteka test.txt?**

```
bash stat -c%h test.txt
```

- **Kaj naredi ukaz chmod 1752 test.txt? Ali se spremeni tudi datoteka test2.txt, ki je trda povezava na datoteko test.txt?**

```
1 111 101 010
```

Sticky bit, uporabnik lahko dela vse, skupina lahko bere in izvaja, ostali pa lahko samo pišejo.

Spremeni se tudi datoteka test2.txt.

- **Kako dobimo pomoč za vgrajene ukaze v Linux-u? Kako preverimo, če je ukaz vgrajen?**

Najprej napišemo `type -a <ukaz>`, tako ugotovimo ali je ukaz vgrajen ali pa ne. Če vrne `<ukaz> is a shell builtin`, to pomeni, da je ukaz vgrajen. V tem primeru dobimo pomoč za ta ukaz `help <ukaz>`. Če ukaz ni vgrajen, pa dobimo pomoč zanj `man <ukaz>`.