

IZPIT 23.6.2011

1. Imamo navidezni pomnilnik z odstranjevanjem. V glavnem pomnilniku imamo prostora le za 4 okvirje. Trenutno so okvirji glavnega pomnika prazni. Pri izvajanju programov pridemo zahteve po podatkih, ki se nahajajo na naslednji straneh navideznega pomnilnika v takšnem vrstnem redu) 3, 37, 10, 4, 4, 3, 45, 1, 2, 4, 10, 5, 5, 64, 4, 1, 3, 5.

- (a) Simulirajte zamenjave strani po principu LRU (najdlje neuporabljen): za vsak dostop narišite sliko glavnega pomnilnika.
 (b) Učinkovitost pristopa primerjajte z optimalno strategijo (na istem primeru), za oba primera izračunajte razmerje zadetkov.
 (c) Opišite, kako deluje optimalna strategija. Kaj je predpogoj, da lahko uporabimo optimalno strategijo?

a)

| | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| 3 | 37 | 10 | 4 | 4 | 3 | 45 | 1 | 2 | 4 | 10 | 5 | 5 | 64 | 4 | 1 | 3 | 5 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 37 | 37 | 37 | 37 | 37 | 45 | 45 | 45 | 45 | 10 | 10 | 10 | 10 | 10 | 1 | 1 | 1 |
| | | 10 | 10 | 10 | 10 | 10 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 3 | 3 |
| | | | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 | 64 | 64 | 64 | 64 | 5 |

10 napak strani, 8/18 zadetkov

b)

| | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|
| 3 | 37 | 10 | 4 | 4 | 3 | 45 | 1 | 2 | 4 | 10 | 5 | 5 | 64 | 4 | 1 | 3 | 5 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| | 37 | 37 | 37 | 37 | 37 | 45 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 64 | 64 | 64 | 3 | 3 |
| | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

5 napak strani, 13/18 zadetkov

c)

- Zamenjaj tisto stran, ki bo potrebna najkasneje
- Lepo, a nemogoče implementirati, saj prihodnosti ne poznam natančno

2. (a) Kaj izpiše funkcija **f**, če jo pokličemo z argumentom 6 (torej **f 6**)? (2)

(b) Kakšen je v tem primeru njen izhodni status? Zakaj? (1)

```
function f {
  if (( $1 <= 0 )); then
    echo 1
  else
    t2=$(f $(( $1 - 1 )) )
    echo $(( $1 * $t2 ))
  fi
}
```

a) Izpis: 6

b) Njen izhodni status je spremenljivka, se pravi 6, ker jo zadnjo izpiše $6 - 5 = 1$ $6 * 1 = 6$ (echo 6).

3. V trenutnem imeniku imamo datoteko skripta.sh, ki ima naslednje pravice: 400 (zapisano v osmiškem zapisu). Želimo, da bi skripto lahko poganjali (vendar pa ne tudi brali in pisali) vsi uporabniki na sistemu.

(a) Napišite ukaz ali zaporedje ukazov s katerimi bi to omogočili. (1)

(b) Kako bi potem nek drug uporabnik v ozadju pognal prej omenjeno skripto, ki se nahaja v njegovem trenutnem imeniku?

- a) `chmod a=xx skripta.sh`
- b) `.\skripta.sh&`

4. Kaj je MUTEX? Kje ga srečamo v okviru operacijskih sistemov? (2)

Semaforje lahko uporabimo pri reševanju problema kritičnih sekcij z več – n procesi. Ideja je v temu, da si procesi delijo semafor mutex (mutual exclusions), ki ima začetno vrednost 1. Uporabljamo ga pri procesih.

5. Naštejte in kratko opišite vsaj 2 prednosti uporabe niti napram kreiranju novega procesa.

- ustvarimo jih hitreje kot proces
- nitko zaustavimo hitreje
- manj časa za preklon med dvema nitkama znotraj istega procesa
- nitke znotraj procesa lahko delijo vire
- nitke lahko komunicirajo brez vpletanja os

6. Kaj je DMA, kako deluje in kje ga uporabljamo? (3)

DMA (Direct memory access) pomeni neposreden dostop procesorja do pomnilnika.

- Procesor pošlje zahtevo DMA kontrolerju
- DMA kontroler nadzira izmenjavo podatkov med glavnim pomnilnikom in V/I napravo
- Procesor je prekinjen le, ko je prenešen celoten blok podatkov

7. Predstavite koncept pomnilniške hierarhije:

(a) Kakšni so razlogi za razvoj pomnilniške hierarhije? (1)

(b) Brez katerega glavnega principa pomnilniška hierarhija ne bi bila možna? Opišite ga! (2)

(c) Skicirajte pomnilniško hierarhijo v tipičnem sodobnem osebem računalniku. (1)

8. (a) Kaj je operacijski sistem? (2)

(b) Napišite vsaj 3 storitve, ki jih nudi OS. (2)

a) Operacijski sistem (kratica OS, angleško *operating system*) je programska oprema nujna za delovanje računalnika. Deluje kot vmesnik med uporabnikom in strojno opremo računalnika.

b)

- Dodeljevanje virov (CPE, pomnilniški prostor, vhodno/izhodne naprave) procesom
- Časovno razporejanje opravil
- Reševanje konfliktnih situacij (sočasni dostopi do virov)
- Optimizacija in nadzira uporabe virov
- Omogočanje dela uporabnikom

9. Opišite koncept mikrojedra! Kakšne so prednosti mikrojedra?

Jedro OS je zelo majhno, vsebuje le bistvene funkcije OS. Veliko storitev, tradicionalno vključenih v OS je sedaj implementiranih v smislu zunanjih podsistemov - strežnikov. Omogoča fleksibilnost in modularno nadgrajevanje.

Prednosti:

Enoten vmesnik za zahteve procesov (vse na podlagi predajanja sporočil).
Razširljivost, fleksibilnost, prenosljivost, zanesljivost, podpora porazdeljenemu sistemu, objektno usmerjeni OS.

10. Naštejte in na kratko opišite 3 pogoje, ki morajo biti izpolnjeni, da obstaja možnost smrtnega objema pri zaseganju virov.

- Vzajemno izključevanje (le en proces lahko uporablja vir v nekem časovnem intervalu)
- drži in čakaj (proces lahko alociran vir drži v lasti in hkrati čaka na dostop do drugega)
- brez sprostitev (procesu ne moremo na silo odvzeti njemu alociranega vira)

11. (a) Naštejte vsaj 5 algoritmov za kratkoročno razvrščanje procesov.

(b) Pri katerih od naštetih algoritmov obstaja nevarnost stradanja procesa? Utemeljite!

- a) Naključno, FIFO, Prioritetno, LIFO, scan
- b) Možnost stradanja je pri LIFO, saj je lahko zahteva zaradi dolgega procesiranja trenutne zahteve potisnjena iz glave vrste (v tem času je torej prišlo več zahtev).

12. Naštejte in na kratko opišite vsaj 4 načine razporejanja posameznih zahtev za dostop do diska.

Zapisi v plasteh (the pile) – Najpreprostejša oblika organiziranja datotek

Podatki so zbrani v vrstnem redu tako kot so prišli; Do datotek lahko dostopamo le z izčrpnim iskanjem skozi celo datoteko.

Zapisi, ki si sledijo v redu (the sequential file) – Najpogostejša oblika organiziranja dat.

Tukaj se uporablja fiksni format za vse zapise, vsi zapisi so iste dolžine, vsi zapisi imajo enako dolga polja v točno določenem vrstnem redu.

Indeksirani zapisi, ki si sledijo v redu (the indexed sequential file)

Način, s katerim obidemo težave organiziranja datotek z zapisi. Datotekam dodamo indekse ter overflow datoteko in s tem dodamo datotekam podporo naključnemu dostopu.

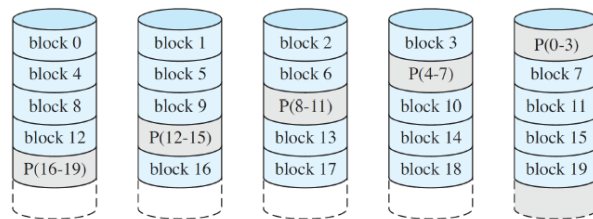
Indeksirani zapisi (the indexed file)

Take zapise se lahko procesira le, ko je znan podatek o ključnem polju. Zato imajo v indeksiranih datotekah vsa različna polja v zapisih svoj indeks, po njem se dostopa.

Datoteka z direktnim dostopom (the direct, or hashed, file)

Ko je potreben zelo hiter dostop do posameznih podatkov v datotekah, pa se uporabi datoteke z direktnim dostopanjem in se lahko dostopa do vsakega bloka direktno.

13. Katera stopnja RAID je prikazana na sliki 1?



Slika 1: RAID; paritetni bloki so označeni s črko P.

Stopnja RAID 3 – pariteta blokov se razporeja z pariteto.

14. Pri sistemih, ki tečejo v realnem času, pravilnost delovanja sistema ni odvisna le od logičnega rezultata procesiranja, ampak tudi od časa, ki je potreben za izračun (dopolnite).

15. Kakšna je razlika med programom in procesom?

Programu, ki se izvaja pravimo proces. Program je algoritem, ki ga lahko izvajamo na računalniku in je napisan v določenem programskem jeziku.

16. Imamo računalniški sistem, na katerem tečejo štirje procesi, ki imajo naslednje pomnilniške zahteve (torej, da se proces v nekem trenutku izvede, potrebuje proces tolikšno količino pomnilnika v celoti):

P1: 30 MB
P2: 60 MB
P3: 40 MB
P4: 100 MB

Trenutno procesi zasedajo sledečo količino pomnilnika:

P1: 20 MB
P2: 20 MB
P3: 5 MB
P4: 10 MB

Procesov ne smemo prekiniti, prav tako ne uporabljamo navideznega pomnilnika. Najmanj koliko MB glavnega pomnilnika mora imeti sistem, da ne bi prišlo do smrtnega objema? Utemeljite! Kakšen bo potem vrstni red izvajanja?

Skupaj zasedejo procesi: $30 + 60 + 40 + 100 = 230\text{MB}$

Trenutno zasedajo procesi: $20 + 20 + 5 + 10 = 55\text{MB}$

Najmanjša velikost pomnilnika je lahko:

$P1(32\text{MB} - 2^5) + P2(64\text{MB} - 2^6) + P3(64\text{MB} - 2^6) + P4(128\text{MB} - 2^7) = 288\text{MB} \rightarrow 512$

$2^9 = 512\text{MB}$ je najmanj gl. pomnilnika, da ne pride do smrtnega objema.

512MB

| | | | | |
|----------|--------------|--------------|--------------|--------------|
| P1(32MB) | Prosto 480MB | | | |
| P1(32MB) | P2(64MB) | Prosto 416MB | | |
| P1(32MB) | P2(64MB) | P3(64MB) | Prosto 352MB | |
| P1(32MB) | P2(64MB) | P3(64MB) | P4(128MB) | Prosto 224MB |