Izpit pri predmetu

OPERACIJSKI SISTEMI

OSVB, FRI, ULJ

18. 9. 2012

Skupaj je možno zbrati **50** točk.

Čas pisanja: **75** minut.

V prvih **10** minutah lahko odstopite od izpita in dobite vrnjeno prijavnico.

V vsakem primeru **MORATE** ta list oddati poleg pole **z imenom, priimkom in vpisno številko**! Slikanje ipd. izpita pred koncem izpita je **prepovedano**! Izpita je konec, ko nadzornik pobere vse izpite. Izpit bo naknadno objavljen na eUčilnici!

Izpit ima 15 vprašanj. Pišite čitljivo!

1. **Naštej vsaj štiri datotečne sisteme? (2)**

Ntfs, fat32, fat16, ext3, ext2

1. **(a) Kakšna je razlika med segmentacijo in ostranjevanjem? (1)**

Ostranjevanje:

• Razdelitev pomnilnika na majhne dele enake velikosti (okvir – frame), hkrati pa razdelitev vsakega procesa v dele iste velikosti (stran – page)

• Zunanje drobitve ni, notranja je prisotna le pri zadnji strani procesa

• Nova podatkovna struktura – OS vzdržuje tabelo strani za vsak proces

– Vsebuje lokacijo okvirja za vsako stran procesa

– Pomnilniški naslov je sestavljen

Segmentacija:

• Segmenti so lahko različnih velikosti

• OS vzdržuje tabelo segmentov za vsak proces

• Naslavljanje: številka segmenta in odmik

• Zaradi neenakosti segmentov je ta tehnika podobna dinamičnemu razdeljevanju; ni notranje drobitve, je pa zunanja, ki pa je manjša zaradi drobljenja procesa na manjše dele, ki niso nujno zvezni v pomnilniku

• Ustreza programerjem: organiziranost kode in podatkov; predstavitev modulov, podatkov za posamezne module

– Podana je maksimalna velikost segmenta, kar programerja žal omejuje

* 1. **Poznaš še kakšno tehniko delitve navideznega pomnilnika? (1)**

Razdeljevanje/Particioniranje (temelji iz starejših sistemov)

1. **Skiciraj in opiši plasti med strojno in uporabniško plastjo. (2)**
2. **(a) Kaj je smisel pomnilniške hierarhije? (2)**

Želimo si hiter dostop, veliko kapaciteto, nizko ceno

* 1. **Kako pomnilniško hierarhijo vidijo aplikacije? (1)**

Vidijo jo kot eno samo celoto.

1. **(a) Razloži kdaj in zakaj prihaja do zunanje drobitve (angl. fragmentation) pomnilnika. (1)**

To so luknje v pomnilniku. Znotraj particije je neizkoriščen prostor. Vsakemu procesu damo nek košček pomnilnika, in ko ga odrabi nastane luknja v pomnilniku. To odpravljamo z algoritmom zasedanja particij.

* 1. **Kateri postopek uporabljamo za reševanje tega problema (kratko ga opišite)? (2)**

Algoritem zasedanja particij.

* 1. **Kaj s tem pridobimo? (1)**

Da so luknje v pomnilniku vedno manjše**.**

1. **(a) Pojasni razliko med monolitnim in mikro jedrom. (2)**

Monolitno jedro vsi servisi OS tečejo v eni sami niti. Monolitno jedro ima bogato

podporo strojni opremi, zato je relativno obsežen. Posledično ga je težje nadzorovati. Razvoj in

implementacija sta lažji kot pri drugih pristopih. Ker se vsi servisi OS nahajajo v istem pomnilniškem prostoru, je izvajanje ucinkovitejše.

Mikro jedro v jedru so implementirani le osnovni servisi kot so upravljanje s procesi,

upravljanje pomnilnika in medprocesno komuniciranje. Ostali servisi tečejo kot programi v uporabniškem načinu imenovanem strežniki. Lažje ga je nadzorovati kot

monolitno jedro, je pa zaradi množice sistemskih klicev delovanje sistema pocasnjejše.

Vsebuje najpogosteje uporabljane in najbolj temeljne komponente operacijskega sistema

* 1. **Kakšno jedro uporablja GNU/Linux? Kakšno Windows 7? (2)**

Linux uporablja monolitno, windows 7 pa mikrojedro**.**

1. **(a) Kaj je razlika med nitjo, procesom in aplikacijo? (3)**

Nit ustvarimo prej kot proces, zaključimo hitreje kot proces, manj časa potrebujemo za preklop 2 nitk znotraj procesa. Nitka je neodvisen PC znotraj procesa. Aplikacija pa je zgrajena iz procesov.

* 1. **Naštejte in kratko opišite vsaj 2 prednosti uporabe niti napram kreiranju novega procesa. (2)**

Nit ustvarimo prej kot proces, zaključimo hitreje kot proces, manj časa potrebujemo za preklop 2 nitk znotraj procesa, ne rabimo vpletati jedro OS saj si nitke znotraj procesa delijo vire.

1. **(a) Smo se z zanko (1)**

**while(TRUE);**

**znašli v smrtnem objemu?**

Ne, za smrtni objem potrebujemo več procesov, ki tekmujejo med seboj, ne pa neskončne zanke.

**(b) Kakšna bo obremenjenost CPE? (1)**

100%, ker je zanka, ki se bo vrtela v neskončnost in bo procesor zmeraj bolj obremenjevala.

1. **(a) Katero lastnost mora imeti optimalen zamenjevalen algoritem strani v navideznem pomnilniku (2) (ali z drugimi besedami kakšen cilj želimo doseči z vsemi zamenjevalnimi algoritmi)?**

Brez napak in čim večja hitrost?

* 1. **Zakaj takšnega algoritma ni mogoče implementirati? (2)**

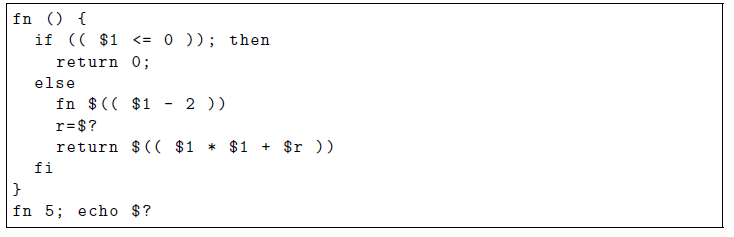
Ker ne moremo doseči enako hitrost kot jo ima procesor?

1. **Ali OS Windows Vista lahko izvaja različne niti enega procesa na različnih procesorjih? (1)**
2. **(a) Naštejte štiri glavne funkcije operacijskega sistema z vidika upravljanja. (2)**

**(b) V katero izmed njih spada koncept trde realne časovnosti? (1)**

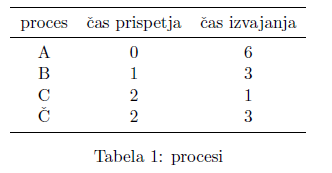
1. **(a) Kaj izpiše spodnja skripta? (3)**

**(b) Kakšen pa je v tem primeru njen izhodni status? Zakaj? (2)**

****

**13. Imamo en procesor in več procesov, ki jih želimo izvajati na procesorju. Čas prispetja in čas**

**izvajanja posameznega procesa je podan v tabeli 1.**

****

1. **Skicirajte kratkoročno razporejanje:**
2. **(a) po kriteriju ''najprej najstarejši" (FIFO { FCFS); (2)**
3. **(b) po kriteriju ''konstantne časovne rezine" (RR). Časovna rezina je dolga 2 časovni enoti, (2)**
4. **čakalna vrsta je ena za vse procese.**
5. **(c) Primerjajte rezultate obeh algoritmov - izračunajte in primerjajte normalizirane čase procesiranja (2)**
6. **(angl. normalized turnaround time). Kateri algoritem je bil v danem primeru boljši?**
7. **14. Na sliki 1 imamo prikazane štiri procese in alociranje virov Ra, Rb, Rc, Rd in Re. Ali pride do (4)**
8. **smrtnega objema? Odgovor obrazložite.**

****

**15. Naštejte in na kratko opišite vsaj 4 načine razporejanja posameznih zahtev za dostop do diska. (3)**