

## Ispit iz predmeta Operativni sistemi 2

Ime i prezime: \_\_\_\_\_

Broj indeksa: \_\_\_\_\_ Broj poena: \_\_\_\_\_/30

*Ispit traje 1,5 sat. Nije dozvoljeno korišćenje literature.*

**1.(3)** Zaokružiti slovo ispred onih karakteristika koje važe za MFQS raspoređivanje procesa u osnovnom/najčešćem obliku.

- a) Za izvršavanje se uzima proces iz prvog sledećeg nepraznog reda u odnosu na onaj iz kog je prethodno uzet proces za izvršavanje (po *round-robin* politici).
- b) Za izvršavanje se uzima proces iz nepraznog reda najvišeg prioriteta, a unutar njega po *round-robin* politici.
- c) Procesu iz reda višeg prioriteta daje se kraći interval vremena izvršavanja.
- d) Procesu iz reda višeg prioriteta daje se duži interval vremena izvršavanja.
- e) Proces koji ponovo postaje spreman nakon I/O operacije smešta se u red nižeg prioriteta od onog iz koga je taj proces prethodno uzet za izvršavanje.
- f) Proces koji ponovo postaje spreman nakon I/O operacije smešta se u red višeg prioriteta od onog iz koga je taj proces prethodno uzet za izvršavanje.

**2.(3)** Na jeziku Java napisati kod za monitor koji realizuje standardni brojački semafor.

Rešenje:

**3.(3)** Da li serverski osluškivač (engl. *listener*) na istoj serverskoj priključnici (engl. *socket*) može uporedo da obavlja komunikaciju sa više klijenata? Ako može, na koji način?

Odgovor:

**4.(3)** U koji model međuprocenke komunikacije (*shared object* ili *message passing*) spada tehnika a) veb servisa (*Web service*), b) razmene podataka preko memorijski preslikanog fajla (*memory-mapped file*)?

Odgovor: a) \_\_\_\_\_ b) \_\_\_\_\_

**5.(3)** U nekom sistemu postoje dva procesa,  $P1$  i  $P2$ , koji koriste resurse  $R1$ ,  $R2$  i  $R3$  na način dat dole. Pretpostavlja se da odmah nakon operacije zauzeća određenih resursa dati proces vrši operacije sa tim resursom, odnosno da je proces ovako konstruisan da bi resurse držao zauzete najmanje moguće (alocira ih najkasnije što može, pre same operacije). Prestrukturirati ove procese tako da se spreči njihova mrtva blokada (*deadlock*) ukidanjem uslova cirkularnog čekanja (*circular wait*). Prestrukturiranje podrazumeva da proces može da zauzme resurs i ranije nego što je to neophodno za operaciju sa tim resursom, ali ne pre nego što je neophodno za sprečavanje mrtve blokade.

$P1$ :	$P2$ :
<code>request(R1);</code>	<code>request(R2);</code>
<code>request(R2, R3);</code>	<code>request(R1,R3);</code>
<code>release(R1,R2,R3);</code>	<code>release(R1,R2,R3);</code>

Rešenje:  $P1$ :  $P2$ :

**6.(3)** Data je sledeća sekvenca referenciranja stranica od strane nekog procesa:

3, 2, 6, 8, 5, 6, 9, 8, 2, 6, 4, 9, 3, 6, 7, 9, 8, 6, 7, 8

Procesu su dodeljena 4 okvira, zamena se vrši lokalno, samo u skupu stranica dodeljenih tom procesu, a inicijalno nije učitana nijedna stranica ovog procesa. Koliko puta ovaj proces generiše straničnu grešku (*page fault*) ako je algoritam zamene stranica LRU?

Odgovor: \_\_\_\_\_

**7.(3)** Koji osnovni problem alokacije memorije rešava *slab* alokator?

Odgovor:

**8.(3)** Na kom disku 0 do 15 se nalazi logički blok broj 67 (blokovi su numerisani počev od 0) kod strukture RAID5 / 4+1 (na svaka 4 bloka 1 blok parnosti)?

Odgovor: \_\_\_\_\_

**9.(3)** Ukratko objasniti šta je .Net CLR.

Odgovor:

**10.(3)** Navesti osnovne slojeve arhitekture Android sistema i kratko navesti šta koji sloj predstavlja-sadrži.

Odgovor: