

OS

Pitanja za ponavljanje

1. Što je to OS?

OS je skup osnovnih programa koji omogućuju provođenje radnih zahvata na računalu.

2. Navesti osnovne dijelove OS-a?

To su: upravljanje datotečnim sustavom, upravljanje spremnikom, upravljanje U/I uređajima, API, GUI, procesi i dretve (komunikacija, sinkronizacija i raspoređivač poslova) i sigurnosni podsustav.

3. Čime su određena svojstva i ponašanje procesora?

Skupom registara i skupom instrukcija.

4. Navesti osnovni skup registara procesora.

Adresni međuregistar, podatkovni međuregistar, programsko brojilo, kazaljka stoga, registar stanja, instrukcijski registar i opći registri.

5. Napisati pseudokod što procesor trajno radi.

```
ponavljati {  
    dohvatiti iz spremnika instrukciju na koju pokazuje programsko brojilo;  
    dekodirati instrukciju, odrediti operaciju koju treba izvesti;  
    povećati sadržaj programskog brojila tako da pokazuje na sljedeću instrukciju;  
    odrediti odakle dolaze operandi i kuda se pohranjuje rezultat;  
    operande dovesti na aritmetičko-logičku jedinku;  
    izvesti zadanu operaciju;  
    pohraniti rezultat u odredište;  
}  
  
dok je (procesor uključen);
```

6. Što je kontekst?

Sadržaj svih registara.

7. Definirati što je program, dretva, proces.

Program je statički niz instrukcija. Dretva je niz instrukcija u izvođenju. Proces je sve ono što je potrebno da bi se program mogao izvoditi, a sastoji se od barem jedne dretve.

8. Kako je moguć višeprogramski rad na jednoprocesorskom računalu?

Tako da se svaka dretva izvodi naizmjenice pa dobijemo privid istovremenosti.

9. Što je radno čekanje?

```
dok je (ZASTAVICA==0); RADNO ČEKANJE  
    čitaj PR;
```

10. Skicirati signale dvožičnog rukovanja.

Pogledaj predavanja slajd 03-16.

11.Što se zbiva kad se dogodi prekid?

a. Što radi CPU kad se dogodi prekid?

Zabranjuje daljnje prekidanje, programsko brojilo stavlja na stog i u programsko brojilo upisuje adresu prve instrukcije prekidnog programa.

b. Što svaka jezgrina funkcija obavlja na početku?

Pohranjuje kontekst.

12.Pojasniti instrukcije pohraniti kontekst i vratiti se iz prekidnog načina.

Pohraniti kontekst znači pohraniti na stog sadržaj svih registara procesora, osim PC-a koji se tretira zasebno. Vratiti se iz prekidnog načina je instrukcija koja vraća sadržaj programskog brojila sa stoga, prebacuje adresiranje u korisnički adresni prostor i aktivira korisnički registar kazaljke stoga.

13. Zašto se programsko brojilo tretira kao zasebno prilikom pohrane konteksta?

Zato da se procesor može vratiti na točno „ono“ mjesto na kojem je bio prekinut.

14.Što se zbiva kada obrada nekog prekida završi?

Vraća se kontekst obrnutim redoslijedom, omogućuje se prekidanje (sa odgodom) i vraća se programsko brojilo.

15.Koje strukture podataka treba sadržavati OS koji omogućuje prihvati prekida različitih prioriteta?

Varijablu T_P (tekući prioritet), polje KON[N] (kontekst pojedine dretve) i polje K_Z[N] (kopije zastavica).

16.Čemu služi sklop za prihvati prekida?

Propušta prema procesoru samo onaj prekidni signal koji ima veći prioritet od dretve koju procesor trenutno izvodi.

17.Opisati prekide generirane unutar procesora?

Pokušaj dijeljenja s nulom, adresiranje nepostojeće lokacije, dekodiranje nepostojeće lokacije.

18.Ulazak u jezgru...

Ulazak u jezgru je moguć jedino prekidom.

19.Navesti osnovne registre pristupnog sklopa za neposredni pristup spremniku.

Podatkovni (PR) i adresni registar (AR), brojilo (BR) te registar stanja (RS).

20. U pseudokodu napisati programski odsječak koji obavlja sklop za neposredni pristup spremniku.

```
dok je( BR > 0 ) {  
    zatražiti sabirnicu;  
    čekati na dodjelu sabirnice;  
    postaviti na adresni dio sabirnice sadržaj registra AR;  
    prenijeti na tu adresu sadržaj podatkovnog registra PR (ili obrnuto);  
    AR = AR + 1;  
    BR = BR - 1;  
}  
postaviti signal PREKID;
```

21. Što je zajedničko procesu roditelju i procesu djetetu?

NIŠTA

22. Koje računalne procese dijele dretve istog procesa?

SVE

23. Kako je podijeljen spremnički prostor procesa a kako dretveni spremnički prostor?

Dretveni prostor je podijeljen na: dio za instrukcije, dio za stog i dio za lokalne podatke dretve. Spremnički prostor se sastoji od više dretvenih prostora i zajedničkog prostora.

24. Navesti uvjet nezavisnosti dretvi .

$(X_i \cap Y_j) \cup (X_j \cap Y_i) \cup (Y_i \cap Y_j)$

25. Max. paralelni sustav zadataka.

Pogledajte Tutoriale!

26. Navesti uvjete koje mora zadovoljavati algoritam međusobnog isključivanja dretvi?

- a) Dretve se odvijaju međusobno isključivo.
- b) Algoritam mora funkcionirati i onda kada su brzine izvođenja različite.
- c) Ako neka dretva zastane u N.K.O. to ne smije spriječiti drugu dretvu da uđe u K.O.
- d) Izbor koja dretva će ući u K.O. mora se zbiti u konačnom vremenu.

27. Za zadani algoritam međusobnog isključivanja ustanoviti jeli ispravan.

To će biti neki pseudokod....

28. Čemu služi Dekerov a čemu Lampardov algoritam?

Dekerov algoritam služi za međusobno isključivanje dvije dretve, a Lampardov za međusobno isključivanje N dretvi.

29. Koje strukture podataka koriste gore navedeni algoritmi?

Deker koristi: varijablu PRAVO i ZASTAVICA, a Lampard koristi: polja BROJ[N] i ULAZ[N] te varijablu ZADNJI_BROJ.

30.Što predstavlja pojam ulazak u jezgru i kada se zbiva?

Pojam predstavlja poziv jezgrine funkcije i zbiva se prekidom.

31.Na što se svodi izlazak iz jezgre?

Izlazak se svodi na pokretanje jedne od dretvi, pri čemu procesor mora biti vraćen u korisnički način rada.

32.Navesti vrste prekida.

Sklopovski prekidi, programski prekidi i prekidi od sata.

33.Od čega se sastoji jezgra?

Od struktura podataka i jezgrinih funkcija.

34.Navesti sadržaj opisnika dretve.

Dvije kazaljke za premještanje iz strukture u strukturu, ID procesa, ID dretve, stanje dretve, prioritet dretve, početna adresa dretvenog adresnog prostora, veličina dretvenog adresnog prostora, adresa prve instrukcije dretve, zadano kašnjenje, prostor za kontekst dretve te programsko brojilo.

35.Navesti strukture podataka jezgre.

....

36.Koja su blokirana stanja dretvi?

Čekanje na binarnom i općem semaforu, čekanje na istek zadanog intervala kašnjenja, čekanje na završetak U/I operacije.

37.Skicirati graf mogućih stanja dretvi.

Skripta predavanja slajd 05-35.

38.Što obavlja instrukcija Aktivirati dretvu iz reda pripravne dretve?

Premješta opisnik dretve iz reda Pripravne_D u red Aktivna_D i nakon toga se iz tog opisnika prenosi kontekst dretve u registre procesora.

39.Što obavlja instrukcija vratiti se iz prekidnog načina?

Vraća u procesor sadržaj programskog brojila i prevodi procesor iz sustavskog u korisnički način rada.

40.Čemu služe jezgrini mehanizmi binarni i opći semafor?

Služe za međusobno isključivanje dretvi.

41.Koje strukture podataka koriste BSEM, OS, OSEM?

BSEM koristi varijablu Bsem[I].v i kazaljku. OS koristi varijablu OS.v i kazaljku.

OSEM koristi varijablu Osem[J].v i kazaljku.

42.U pseudokodu napisati jezgrine funkcije Čekaj_BSEM (ili OS, ili OSEM) i

Postavi_BSEM(ili OS, ili OSEM)?

Ima u skripti za predavanja. Neda mi se prepisivati to sve. To bude vrlo vjerojatno jer je Golub to naglasio.

43.Sinkronizirati sustav dretvi nekog sustava zadataka.

Opet Tutorial...

44. Koje vrste prekida uzrokuju jezgrine funkcije Započeti U/I i Prekid U/I?

Sklopovski prekid.

45. Može li se kontekst dretve koja obavlja neku jezgrinu funkciju naći u opisniku AktivnaD?

Ne, jezgrina funkcija ne može biti prekinuta.