

AROS – drugi deo

1. Kako je E. Dijkstra rešio problem međusobnog isključenja i sinhronizacije procesa?

E. Dijkstra je dao rešenje problema međusobnog isključenja i sinhronizacije procesa tako što je uveo novi tip promenljive koja se naziva semfor i čiji je skup dozvoljenih vrednosti skup nenegativnih celih brojeva. Pritom je uveo dve dozvoljene operacije nad promenljivim (P i V).

2. Prikazati rešenje problema međusobnog isključenja $n \geq 2$ procesa korišćenjem mediska.

Za više procesa koji imaju kritične sekcije jedan u odnosu na druge koristi se promenljiva medisk(međusobno isključenje) koja je tipa semafor.

```
var medisk: semaphore;
```

```
begin
```

```
    medisk:=1;
```

```
    parbegin
```

```
        Pi : begin
```

```
            repeat
```

```
                P(medisk);
```

```
                kritična_sekcija_i;
```

```
                V(medisk);
```

```
                ostale_naredbe_i;
```

```
            until stop;
```

```
        end;
```

```
    parend;
```

```
end;
```

3. Definisati operacije P i V u radu sa semaforima.

$P(s) = s: = s-1$ ako je $s > 0$

$P(s) = \text{čekaj}$ ako je suprotno

$V(s) = s: = s+1$

gde je s promenljiva tipa semafor

4. Koje vrste organizacije virtuelne memorije postoje?

- segmentna
- stranična
- segmentno-stranična

5. Šta je stranični prekid i kako se servisira?

Kada se u toku izvršavanja programa traži pristup adresi koja pripada strani koja nije u memoriji dolazi do prekida programa. Ova vrsta prekida naziva se stranični prekid.

6. Objasniti lokalno i globalno straničenje.

Globalno straničenje – ukoliko su nakon straničnog prekida svi raspoloživi okviri kandidati za izbacivanje.

Lokalno straničenje – ukoliko su samo stranice u okvirima koje pripadaju datom procesu kandidati.

7. Metode dodele prostora na disku koje koriste šeme sa indeksima, svakoj datoteci pridružuju tabelu indeksa. Koja su moguća rešenja za dodelu prostora na disku tabeli indeksa?

Kao za obične datoteke:

- dodela susedne memorijske lokacije
- povezana lista blokova
- korišćenje indeksa u više nivoa

8. Objasniti SCAN algoritam.

Ovaj algoritam razrešava problem "gladovanja" koji se može javiti kod SSTF algoritma. Problem se razrešava tako što se nakon pokretanja glave diska dalje pokretanje nastavlja u istom smeru, tj najpe se servisiraju zahtevi počev od najdalje spoljašnje staze ka unutrašnjoj stazi koja je krajnja, a zatim od najbliže unutrašnje ka najdaljoj spoljašnjoj.

9. Koje funkcije preslikavanja postoje kod upravljanja memorijom?

Tipovi preslikavanja kod upravljanja memorijom su:

- preslikavanje imena
- preslikavanje adresa
- preslikavanje sadržaja

10. Šta su logičke, a šta primarne particije?

Primarne particije su one sa kojih je moguće podizanje operativnog sistema. Svaki disk mora da ima bar jednu primarnu particiju. Upotrebom više primarnih particija moguće je instalirati i koristiti više operativnih sistema na istom disku.

Logičke particije su particije čija je namena skladištenje podataka. Sa logičkih particija se ne može podizati operativni sistem.

11. Neka je dat bafer kapaciteta C poruka, proces S koji upisuje poruke u bafer i proces R koji čita poruke iz bafera. Neka je broj poruka u baferu označen sa N . Koji uslovi moraju biti ispunjeni da bi sistem ispravno funkcionisao?

Da bi procesi S i R ispravno funkcionisali, u situaciji kada njihove relativne brzine nisu unapred poznate, neophodno je da budu ispunjeni sledeći uslovi:

- 1) pošiljalac S ne sme da šalje poruke kada je bafer pun, jer bi se ona izgubila
- 2) primalac R ne sme da uzima poruku kada je bafer prazan, jer bi to mogao protumačiti kao signal da treba da prestane sa radom
- 3) procesi ne smeju istovremeno pristupiti baferu
- 4) vrednost promenljive N mora, u trenutku pristupa baferu bilo kog procesa, biti jednaka stvarnom broju poruka u baferu

12. U kom delu računara se nalaze procesi u stanju: 1) Blokiran/Suspendovan 2) Spreman/Suspendovan? Koja je razlika između ova dva stanja procesa?

1) Proces je u spoljnoj memoriji i čeka na događaj

2) Proces je u spoljnoj memoriji, ne čeka na događaje, ali će postati spreman za izvršavanje kada ga operativni sistem vrati u memoriju

Proces koji je suspendovan, prestaje da se takmiči za resurse, oslobađaju se resursi koje je zauzeo, ali ostaje i dalje proces. Proces koji je u stanju suspendovan/blokiran prelazi u stanje suspendovan/spreman, ako postaje spreman, tj. ako može da nastavi sa radom.

13. Dat sistem ima p procesa. Svakom procesu je potrebno najviše m resursa datog tipa. Ukupno je raspoloživo r resursa. Koji uslov mora biti ispunjen da se ne bi pojavio deadlock?

$$r \geq p(m - 1) + 1$$

r -raspoloživi resursi, p -broj resursa, m -potreban broj resursa

14. Šta je inverzija prioriteta procesa? Objasniti kako nastaje.

Ako proces višeg prioriteta zahteva pristup resursu koji je zauzet od procesa nižeg prioriteta, tada je proces višeg prioriteta blokiran procesom nižeg prioriteta, sve dok proces nižeg prioriteta ne završi rad i oslobodi resurs. Na taj način proces nižeg prioriteta ima "viši" prioritet.

15. Kako se može rešiti problem inverzije prioriteta procesa?

Proces nižeg prioriteta koji pristupa datom resursu nasleđuje viši prioritet procesa koji zahteva pristup istom resursu, sve do završetka rada sa resursom. Na taj način se ubrzava izvršavanje procesa nižeg prioriteta. (Drugi procesi možda imaju veći prioritet od pomenutog procesa, pa proces višeg prioriteta, jer je njemu potreban resurs zauzet, može beskonačno da čeka) Kada proces nižeg prioriteta završi, njegov prioritet se vraća na originalnu vrednost. To je protokol nasleđivanja prioriteta.

16. Objasniti šta je „trashing“.

Stanje sistema kada se mnogo više vremena troši na obradu straničnih prekida nego na izvršavanje procesa. Rešava se smanjenjem stepena multiprogramiranja.

17. Navesti koje vrste planera postoje kod operativnih sistema i objasniti njihovu namenu.

Dugoročni planer - selektuje procese koji su poslani na izvršavanje i prebacuje ih u red procesa spremnih za izvršavanje.

Kratkoročni planer - selektuje proces iz reda procesa spremnih za izvršavanje i dodeljuje mu centralni procesor. Kratkoročni planer se veoma često poziva za razliku od dugoročnog.

18. Šta je deskriptor procesa (ili kontrolni blok procesa – KBP) i koje podatke sadrži?

Kontrolni blok procesa (KBP) je skup podataka o aktivnom procesu o kojima operativni sistem vodi računa. Ti podaci su: ime procesa, identitet vlasnika, prioritet, PSW, oblast programa, oblast podataka, vrednost registara, logičko stanje itd.

19. Navesti koja je osnovna razlika između Ext3 i Ext2 sistema datoteka.

Ext3 je proširenje sistema Ext2 i koristi se u Debian, Red Hat i drugim distribucijama Linux-a. Osnovna razlika između Ext2 i Ext3 je to što je dodat dnevnik transakcija.

20. Navesti kriterijume koji se koriste kod algoritama planiranja.

1. Iskorišćenje CPU-a
2. Propusnost
3. Vreme provedeno u sistemu (turnaround time)
4. Vreme čekanja
5. Vreme odziva

21. U uslovima višeprogramskog rada može doći do situacije koja se naziva potpuni zastoj ili blokiranje ("deadlock"). Navesti potrebne uslove za nastanak potpunog zastoja.

- Međusobno isključenje ("mutual exclusion")
- Posedovanje i čekanje ("hold-and-wait")
- Kružno čekanje ("circular wait")
- Nema prekidanja ("no preemption")

22. Šta je tabela deskriptora sistema datoteka?

Ove tabele sadrže pokazivače na sve datoteke koje je dati proces otvorio.

23. Koje se od sledećih komponenata stanja programa u toku izvršavanja dele između niti datog multithreaded procesa:

- a) vrednosti registara
- b) heap memorija
- c) globalne promenljive
- d) stack memorija?

Treba zaokružiti heap memoriju i globalne promenljive.

24. Objasniti HPF algoritam dodele procesora.

HPF je planiranje po prioritetu. Svaki proces ima pridružen odgovarajući prioritet. Naredni proces za izvršavanje je onaj proces koji ima najviši prioritet.

25. Većina sistema omogućava programima da u vreme izvršavanja alociraju dodatni memorijski prostor. Šta je potrebno za podršku dinamičke memorijske alokacije u sledećim slučajevima:

- a) kontinualna alokacija memorije
- b) upravljanje pomocu segmenata
- c) upravljanje pomocu stranica

- a) kontinualna alokacija memorije - relokacija celog programa
- b) upravljanje pomoću segmenata- relokacija celog segmenta
- c) upravljanje pomoću stranica- inkrementalna alokacija novih stranica (moguća bez potrebe relokacije adresnog prostora programa)

26. Navesti probleme koji postoje kod softverske realizacije algoritama za upravljanje kritičnim sekcijama.

Stalno testiranje promenljivih ili stanja čekanja, svi detalji implementacije direktno zavise od programera, ne postoji način da se nametne protokol koji zavisi od kooperacije, ovi protokoli su suviše komplikovani.

27. U kojim slučajevima se za komunikaciju između 2 procesa koristi deljena memorija (shared memory) umesto razmene poruka (message passing)? Objasni odgovor.

Razmena poruka je pogodna za manju količinu podataka jer nema opasnosti od konflikata. Takođe, ona se lakše implementira od deljenje memorije.

Deljena memorija omogućava brže komuniciranje (zbog blizine memorije procesorima), ali zahteva da se obrati posebna pažnja na zaštitu i sinhronizaciju procesa koji je koriste. Takođe, bolje koristi konvencionalnu arhitekturu jer se postojeći procesori mogu lako dodati u deljeni system.

-Ostala pitanja-

1. Objasniti šta znači potpuni zastoј ("deadlock")?

To je situacija kada proces trajno ostaje u stanju čekanja, jer u isto vreme postoje drugi procesi koji su takođe u stanju čekanja, a zauzeli su resurse koje zahteva dati proces i te resurse drže zauzetim. U takvoj situaciji dati proces nikada ne dobija zahtevane resurse.

2. Koja su moguća stanja procesa?

- 1) NOV – proces je kreiran
- 2) IZVRŠAVA SE – instrukcije datog programa se izvršavaju
- 3) ČEKA – proces čeka da se neki događaj dogodi
- 4) SPREMAN – proces čeka da bude dodeljen procesoru
- 5) ZAVRŠEN – proces je završio izvršavanje

5. Objasniti prelaske između mogućih stanja procesa.

Nakon kreiranja procesa, proces se nalazi u stanju NOV. Kada se procesu dodele potrebni resursi, osim procesora, tada on prelazi u stanje SPREMAN. Nakon dodele procesora prelazi u stanje IZVRŠAVA SE. Po isteku dodeljenog kvantuma vremena proces prelazi iz stanja IZVRŠAVA SE u stanje SPREMAN. Ako u stanju IZVRŠAVA SE mora da sačeka neki događaj (npr. završetak neke U/I aktivnosti), tada proces prelazi u stanje ČEKA i o tome obaveštava operativni sistem. Po završetku U/I aktivnosti proces iz stanja ČEKA prelazi u stanje SPREMAN. Proces prelazi iz stanja SPREMAN u stanje IZVRŠAVA SE kada procesor postane raspoloživ. Ako se u toku izvršavanja procesa završi izvršavanje svih programski instrukcija tada proces prelazi u stanje ZAVRŠEN.

3. Vrste algoritama dodele procesora.

- FCFS (First Come First Served)
- SJF (Shortest Job First)
- Priority Scheduling
- HPF (Highest Priority First)
- RR (Round-robin)
- SRTF (Shortest Remaining Time First)
- MLS (Multilevel Strategy)

4. Algoritmi za dodelu diska.

- FCFS - najjednostavniji algoritam, izvršava zahteve za dodelu diska onim redosledom kojim su nastali

- SSTF - algoritam koji u datom trenutku vrši pokretanje glave diska uzevši u obzir zadate U/I zahteve, osnovna ideja je minimiziranje vremena pozicioniranja u odnosu na trenutnu poziciju glave diska

- SCAN

- C-SCAN – algoritam koji omogućuje uniformnija vremena odziva tako što se pokretanje glave diska uvek vrši u jednom smeru
- LOOK – algoritam koji modifikuje SCAN algoritam, tako što zaustavlja kretanje glave u istom smeru ako nema više U/I zahteva u tom smeru
- C-LOOK – algoritam koji servisira zahteve pomeranjem glave diska od spoljašnjeg ka unutrašnjem cilindru dokle god ima zahteva, a zatim se vraća na spoljašnji najudaljeniji cilindar za koji postoji zahtev

5. Šta su "stack" i "heap" i čemu služe?

Stack memorija služi za memorisanje parametara prilikom poziva procedura i za lokalne promenljive. Drugim rečima, za podatke koji se dinamički dodeljuju po unapred definisanom redosledu(predictable order). Ovom memorijom upravlja kompajler, a ne korisnik.

Heap memorija se koristi za proizvoljne strukture podataka, kao što su LinkedList itd. Drugim rečima, za podatke koji se dinamički dodeljuju na način koji nije unapred definisan(unpredictable manner). Ovom memorijom upravlja korisnik.

6. Objasniti šta je zaposleno čekanje ("busy wait").

Zaposleno čekanje je situacija kada proces koji pokušava da izvrši P operaciju u situaciji kada je vrednost promenljive s jednaka nuli, troši vreme centralnog procesora, pri čemu ne može da napreduje dalje, a istovremeno onemogućava ostale procese da se izvršavaju.

7. Na koji način se operativni sistem "budi" iz stanja praznog hoda?

- prekidom dobijenim od nekog hardverskog uređaja
- pojavom izuzetka od nekog korisničkog programa
- sistemskim pozivom iz nekog korisničkog programa

8. Opisati "Best Fit" algoritam (algoritam najboljeg uklapanja).

Kod "Best Fit" algoritma tabela slobodnog prostora uređuje se po rastućoj veličini slobodnog prostora. Ovaj algoritam se svodi na nalaženje najmanjeg dovoljno velikog slobodnog memorijskog prostora u koji može da se smesti novi program.

9. Opisati "First Fit" algoritam (algoritam prvog uklapanja).

Kod "First Fit" algoritma operativni sistem vodi tabelu slobodnog prostora uređenu po rastućim adresama slobodnih delova memorije. Svodi se na pronalaženje prvog slobodnog prostora, računajući od početka memorije, koji je dovoljno veliki da prihvati dati program.

10. Kolika je maksimalna veličina datoteka i particija kod FAT16 i FAT32?

Kod FAT16 – maksimalna veličina datoteke i particije je 2GB. Kod FAT32 – maksimalna veličina datoteke je 4GB, a particije 2TB.