

Nekoliko zadataka s ispitnih rokova iz predmeta

OPERACIJSKI SUSTAVI

1. U nekom mp3 uređaju s zaslonom generiraju se dva tipa prekida. Prekid P1 koji se generira svakih 10 [ms] uzrokuje dekodiranje jednog segmenta glazbe i slanje tog segmenta zvučnom podsustavu. Trajanje obrade tog prekida jest 2 [ms]. Drugi prekid P2, manjeg prioriteta, javlja se svakih 40 [ms] i pokreće izračun slike koja se prikazuje na zaslonu uređaja. Trajanje izračuna iznosi 10ms. Prekid većeg prioriteta prekida obradu prekida manjeg prioriteta. Procedura za određivanje prioriteta prekida traje 0,1 [ms]. Može li se na uređaj staviti i program koji zahtijeva oko 45 ms procesorskog vremena (može biti prekidan) svakih 200 ms? Obrazložiti.
2. U nekom kinu postoji više šaltera za prodaju karata, a postoji i mogućnost rezervacija preko Weba. Svaka aplikacija koja prodaje ili rezervira karte pri radu se spaja na centralni poslužitelj koji za svaki zahtjev stvara novu dretvu koja obavlja rezervaciju (ili prodaju) funkcijom `obradi_zah_tjev()`. Ako kino ima 250 sjedala predložite rješenje centralnog sustava, tj. kod koji svaka dretva radi i strukturu podataka koje one koriste. Pretpostaviti da dretva čita zahtjev za rezervaciju funkcijom:

```
procitaj_zah_tjev(sjedalo, br_karata),  
    //sjedalo - redni broj prvog sjedala koje se želi rezervirati,  
    //br_karata - broj karata, počevši od 'sjedala',
```

te odgovara sa:

```
salji_odgovor(status),  
    //status = 1, rezervacija uspjela,  
    //status = 0, rezervacija nije uspjela (netko je drugi  
    //rezervirao neko od traženih mjesta).
```

Sinkronizaciju dretvi obaviti monitorom. (Detalji tko je rezervirao nam nisu bitni; pretpostaviti da postoji samo jedna projekcija za koju se program radi!)
3. Riješiti problem sinkronizacije dretvi pisača i čitača (napisati pseudo kod funkcija *pisač()* i *čitač()*) korištenjem monitora te dodatnih varijabli (ako je potrebno). Pretpostaviti da pisači u kritičnom odsječku *pišu* funkcijom *piši()*, a čitači čitaju funkcijom *čitaj()*. Dretve pisača i čitača se proizvoljnim redoslijedom javljaju u sustavu. Problem se sastoji u tome da dok neki čitač čita ni jedan pisač ne smije pisati, dok istovremeno drugi čitači mogu čitati. Dok neki od pisača piše svi ostali moraju čekati. Dati prednost pisačima, tj. čim se pojavi pisač svim novim čitačima zabraniti ulaz u kritični odsječak dok ima pisača koji čekaju. Radno čekanje nije dozvoljeno!
4. Za poslužitelje P1 i P2 istih karakteristika koji obrađuju poslove istih svojstava, izmjereno je da prosječno rade s opterećenjem od 20% (faktor iskorištenja). Kada bi P1 prihvatao i poslove od P2, koliko bi se povećalo prosječno vrijeme zadržavanja posla u sustavu P1 (izraziti u omjeru ili postocima)?
5. Za jedan poslužitelj P1 izmjereno je da prosječno radi s opterećenjem od 90% (faktor iskorištenja). Kada bi polovicu posla koje dolazi na poslužitelj P1 preusmjerili na poslužitelj P2, koliko bi se smanjilo prosječno vrijeme zadržavanja posla u sustavu P1 (izraziti u omjeru ili postocima)?
6. Intelov sklop za podršku straničenja pri radu koristi hijerarhijsko 2-razinsko stablo za pohranu opisnika stranica pojedinog procesa. Zbog toga bi za svaku pretvorbu logičke adrese trebao dva puta dohvatiti sadržaj iz memorije. Na koji je način je riješen ovaj problem (da se smanji broj dohvata koje radi MMU)?
7. Matrice A i B dimenzija $(2N) \times (2N)$ riječi sadrže neke podatke (veličina okvira je N riječi). Ukoliko program treba zbrojiti te matrice i rezultat pohraniti u matricu C (istih dimenzija), odrediti koliko će promašaja izazvati program, ako u radnom spremniku za podatke programa (matrice A, B i C) postoji:
a) 1 okvir, b) 6 okvira.

Algoritam zbrajanja je:

```

za i=1 do 2N radi {
    za j=1 do 2N radi {
        C[i, j] = A[i, j] + B[i, j];
    }
}

```

8. U sustavu sa straničenjem veličina stranice je **4M** riječi, a one se pune na zahtjev. Algoritam zamjene stranica je LRU. Matrica **A**[1..N, 1..M] je pohranjena po recima (na

```

za i = 1 do N-1 {
    za j = 1 do M {
        A(i+1, j) = A(i, j) + A(i+1, j);
    }
}

```

susjednim mjestima mijenja se desni indeks). Koliko promašaja će izazvati prikazani algoritam, ako za matricu **A** u radnom spremniku postoji: a) jedna stranica, b) dvije stranice. (Zanemariti promašaje zbog dohvata naredbi samog programa i pristupa pomoćnim varijablama.)

9. Zadan je algoritam za množenje kvadratnih matrica:

```

for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
        x = 0;
        for (k = 0; k < n; k++)
            x += a[i][k] * b[k][j];
        c[i][j] = x;
    }

```

Neka je u nekom sustavu sa straničenjem veličina stranice 8 riječi, a matrice koje se množe su dimenzija 4x4 (svaka matrica stane u 2 stranice). Napisati niz zahtjeva za podatke u stranicama koje će generirati navedeni program na zadanom primjeru. Koliko će promašaja izazvati računanje jednog retka izlazne matrice ako se primjeni LRU strategija uz tri raspoloživa okvira za program?

10. Uzimajući u obzir performanse, isplati li se povećati brzinu okretanja diskova s 7200 na 10000 ili se više isplati prijeći s 2,5" na 2" diskove, uz isti broj staza po inchu, ali 25% više sektora po stazi, uz pretpostavku da mehanički dio i dalje ima ista svojstva, tj. brzine pomaka glave? Za usporedbu koristiti čitanje kompaktno smještene datoteke koja bi na 2" disku zauzimala 16 staza, a za postavljanje glave koristiti prosječno vrijeme traženja (trećina svih staza). Diskovi imaju (u obje inačice) četiri magnetske ploče, a pomicanje glave (za veće pomake) za OBA diska može se izraziti formulom: $T = 6 + 6 * D / \text{MAX}$ [ms] (MAX – ukupan broj staza za 2,5" disk). Zanimati trajanje premještanja glave sa staze na stazu kao i premještanje pročitanih sektora u glavnu memoriju.
11. Proizvođači tvrdih diskova da bi povećali kapacitet diskova imaju dvije opcije (koje ih jednako koštaju i izvedive su) ili povećati broj magnetskih ploča ili povećati gustoću zapisa (više staza i više sektora po stazi). S obzirom na ukupne performanse diska usporediti ove dvije opcije.
12. U NTFS datotečnom sustavu, datoteka veličine 10 MB pohranjena je po dijelovima u uzastopnim blokovima veličine 4 KB, kako slijedi: prvih 4 MB počevši od bloka (LCN) 34567, slijedeća 2 MB počevši od bloka 12345, te zadnjih 4 MB počevši od bloka 20000. Napisati sažeti zapis opisa smještaja (VCN, LCN, #).
13. Veličina bloka podataka s kojom je napravljena datotečna tablica na UNIXu je 1 kB, a adrese su 32-bitne. Grafički prikazati opisnik datoteke te dodatne kazaljke (i njihovu organizaciju) za datoteku veličine 10 MB.