



TENTAMEN I OPERATIVSYSTEM, HI1025:TEN1 - 20 MARS, 2015

Allmänna instruktioner. Tentamen innehåller 10 frågor med totalt 46 poäng. För lägsta godkända betyg (E) krävs ungefär 23 poäng. För att få komplettera (Fx) krävs ungefär 21 poäng.

Viktigt: När du svarar på en fråga ska det i allmänhet framgå varför du vet svaret frågan, det uppnås enklast genom att du sätter in termer och begrepp som du hanterar i deras sammanhang på ett korrekt sätt. Men det är också viktigt att inte bli för mångordig, i uppgifter med 1 poäng krävs ett kort svar, i uppgifter med mer poäng behöver svaret utvecklas mer. **OBS: Svara *inte* i tentan, *bara* på svarspapper.**

Anmärkning: Av layoutskäl saknas nödvändiga #include-direktiv i källkoden i vissa av uppgifterna, men uppgifterna ska behandlas som om direktiven fanns där.

Lycka till!

Johnny

UPPGIFTER

- (1) Förklara vad *mutual exclusion/ömsesidig uteslutning* är och hur det implementeras. Din redogörelse ska inkludera och korrekt behandla begreppet *race condition/kapplöpning* samt förklara varför problemet måste få sin lösning på maskindesignnivå. (Vi behöver alltså göra något speciellt på låg nivå - maskinnivå. Vad? Varför?)
(4p)
- (2) Redogör för vad en så kallad *context switch* är mellan processer och förklara varför det är kostsamt att genomföra en context switch.
(4p)
- (3) Förklara följande begrepp / instrument
 - (a) *DMA/Direct Memory Access*
 - (b) *Round-robin scheduling*
 - (c) *FIFO scheduling*
 - (d) *MMU/Memory Management Unit*
(8p)
- (4) Förklara skillnaden mellan primärminne och sekundärminne och förklara varför primärminnet jämfört med sekundärminnet har en lägre lagringskapacitet men en kortare åtkomsttid.
(4p)
- (5) Då man vill att ett program ska kunna ha parallella aktiviteter så har man två alternativ, antingen förlägga dessa parallella aktiviteter i parallella trådar inom en och samma process eller skapa enskilda parallella processer, en för varje parallell aktivitet. Diskutera för- och nackdelar med båda valen på detta sätt: Ange två fördelar med varje val som också är nackdelar med det andra valet. Du ska också förklara varför varje fördel är en fördel och varför den blir en nackdel för det andra valet. (Du skriver alltså ungefär: *trådar är bra för att bla bla bla och detta är inte bra med processer för att bla bla bla* osv med tre fördelar till, en för trådalternativet och två för processalternativet.) (4p)
- (6) Förklara begreppet *thrashing* och ange vad man gör för att undvika det. I din text ska begreppen *sidfel* och *swapping* ingå och vara korrekt hanterade. (4p)
- (7) Det finns ett antal olika tillstånd som en process kan befinna sig i. Förklara följande tre olika tillstånd: *Running, Sleeping, Zombie/Defunct*. Din text ska inte vara en översättning av de olika engelska orden utan förklara betydelsen i de olika tillstånden, till exempel vad har en process i tillståndet *Zombie/Defunct* för egenskaper som skiljer den från en process i de andra tillstånden? (3p)

(8) Hur används avbrott för att möjliggöra timesharing av CPU:n? Förklara detta genom att konkret beskriva hur avbrottshanteringen kommer in då flera processer som vill ha processortid får det. (4p)

(9) Studera nedanstående program:

```
main(){
    int x=1,y=2,z=0,u=0,v=0,w=0, p1[2], p2[2]; pipe(p1); pipe(p2);
    if(!fork()){
        if(!fork()){
            close(p1[1]); close(p2[0]);
            while(read(p1[0],&u,sizeof(int)))w=w+u;
            write(p2[1],&w,sizeof(int));
            close(p1[0]); close(p2[1]);
            exit(0);
        }
        wait(0);
        exit(0);
    }
    close(p1[0]); close(p2[1]);
    write(p1[1],&x,sizeof(int)); write(p2[1],&y,sizeof(int));
    wait(0);
    read(p2[0],&z,sizeof(int)); printf("Summan: %d\n", z);
    close(p1[1]);close(p2[0]);
}
```

- (a) Programmet fungerar inte som det ska, beskriv programmets funktion och rita ett tidsdiagram som illustrerar de ingående processernas relationer och rita även in de två piparna. Förklara vad som händer när programmet kör och ange dess utmatning.
- (b) Programmet ska summera de två talen 1 och 2 genom att göra detta i en barnbarnprocess, men, som sagt, det fungerar inte, rätta programmet så att det fungerar som det ska och förklara varför din rättning medför att programmet fungerar. **OBS:** Koden i barnbarnet får inte ändras.

(7p)

(10) Studera nedanstående program:

```
int p[2]; char ch='*';
void* tf (void * q){
    pthread_t t1,t2 = 0; int *t = (int *) q; int next;
    if(q>0){
        write(p[1],&ch,sizeof(char));
        next = *t-1;
        if(next>0){
            pthread_create(&t1,NULL,&tf,&next);
            pthread_create(&t2,NULL,&tf,&next);
            pthread_join(t1,NULL);
            pthread_join(t2,NULL);
        }
    }
    sleep(2);
    close(p[1]);
}
main(){
    pthread_t t; char ch2; pipe(p); int first = 4;
    pthread_create(&t,NULL,&tf,&first);
    sleep(1);
    while(read(p[0],&ch2,sizeof(char)))printf("%c\n", ch2);
    pthread_join(t,NULL);
}
```

Vad blir utskriften av programmet? Motivera ditt svar genom att förklara programmets körning.

(4p)