

Tentamen i 2G1504 (fd 2G1113) Operativsystem

torsdag 2004-08-19 kl 0900-1300

Institutionen för Mikroelektronik och Informationsteknik

Examinator: Robert Rönngren

Hjälpmedel: Inga

Tentamensfrågorna behöver inte återlämnas efter avslutad tentamen.

Ange på omslaget vilken kurs du tenterar (2G1113 eller 2G1504) och vilken termin du läste kursen.

Varje inlämnat blad skall förses med följande information:

- Namn och personnummer
- Nummer för behandlade uppgifter
- Sidnummer

Rättning:

- **Alla svar**, även på delfrågor, måste ha åtminstone en kortfattad motivering för att ge poäng
- Resultat beräknas anslås senast mån 13/9

Betygsgränser:

- Totalt antal poäng på tentamen är 31
- Godkänt (3) garanteras från 16 poäng

Lösningsförslag: anslås på kursens webbsida efter tentamen (antagligen inte förrän 20/8)

Frågor: (som normalt bara kräver kortfattade svar 1-10 rader)

- 1) Till en process finns knutet ett process control block (PCB), Informationen i PCB kan generellt delas in i tre kategorier - vilka? (1.5p)
- 2) Rita en schematisk bild av adressöversättningen i ett system med sidindelning och TLB och en-nivåsidtabell. (1p)
- 3) I ett system finns både korta jobb som i medeltal exekverar 0.1 sek innan de blockerar på I/O eller är klara, och långa jobb som i medeltal exekverar 10sek-10min. Diskutera för- och nackdelar med att använda schemalägningsstrategierna FCFS, SJF respektive round-robin för systemet (3p)
- 4) I schemalägningsystem talas ofta om måttet "throughput". Vad innebär det? (1p)
- 5) Kan en vanlig användare påverka schemaläggningen i LINUX/UNIX? Hur? (1p)
- 6) I vilken del av en modern datorarkitektur implementeras normalt adressöversättningen i det fall man har virtuelltminne baserat på sidindelning? (1p)
- 7) Hur bör man dela upp en virtuell adress för att minimera utrymmet som krävs för en två-nivåers sidtabell givet en virtuell adressrymd om 2^{32} , en fysisk adressrymd om 1GB, sidstorlek på 4KB, lika många bitar (β) används för en post (entry) i den yttre som inre sidtabellen, och att en genomsnittlig process använder 16MB för stack, 8MB data och 8MB för text? (2p)
- 8) Beskriv begreppen dynamisk laddning respektive dynamisk länkning! (2p)
- 9) Hur brukar man säkerställa att en användare inte kan exekvera en fil med t.ex data eller text även om den har rättigheter i filsystemet (protection) som medger exekvering? (1p)
- 10) i) I filsystemssammanhang används ofta en "block cache". Vad är det? (1p)
ii) Varför vill man inte ha ren LRU som utbytesstrategi för block cachén? (1p)
- 11) Kalle hävdar att context switch tiden att byta mellan två kärntrådar är betydligt mindre än att byta mellan två processer - har han rätt eller fel? Förklara någorlunda utförligt varför? (2p)

- 12) Antag att vi har ett system med sidstorlek som rymmer 500 heltal av typen `int`. Systemet har ett primärminne med 3 ramar. Nedanstående programs textarea rymmer i en ram. Hur många sidfel genereras, givet att matriser lagras radvis och sidutbytesalgoritmen är LRU, när respektive program i och ii exekveras om vi har ren demand-paging? (2p)

Program i	Program ii
<pre>void main(int argc, char **argv) { int arr[50][100]; int i, j; for(i=0; i<50; i++) for(j=0; j<100; j++) arr[i][j] = 0; }</pre>	<pre>void main(int argc, char **argv) { int arr[50][100]; int i, j; for(i=0; i<100; i++) for(j=0; j<50; j++) arr[j][i] = 0; }</pre>

- 13) Kalle skall designa ett lastbalanseringsystem för ett multiprocessor system. Han brottas med en grundläggande designfråga där han har två val:
 i) En CPU som är överlastad försöker lämna jobb till sina grannar för att minska sin last
 ii) En CPU som är underlastad frågar sina grannar om jobb att utföra
 Vilken strategi bör Kalle välja? (1p)
- 14) Beskriv stegen i ett remote procedure call! (1.5p)
- 15) Bankir algoritmen kan användas för att undvika deadlock genom att hålla ett system i "säkert" tillstånd. Givet följande system, vad är det minsta värdet på X som gör att systemet är i ett säkert tillstånd? (1p)

Process	Allokerat			Max behov			Tillgängligt		
A	1	2	3	6	9	4	2	X	2
B	0	4	2	3	7	5			
C	1	1	4	3	2	4			
D	2	2	3	5	8	9			

- 16) I säkerhets sammanhang nämns ofta följande begrepp/namn:
 i) CERT
 ii) Orange book
 Beskriv dem kortfattat! (2p)
- 17) i) Beskriv det vanligaste sättet att skapa digitala signaturer. (2p)
 ii) Vilket skydd ger det användaren av en applet om den är försedd med en digital signatur? (1p)
- 18) När är ett periodiskt realtidsystem schemalägningsbart? (1p)
- 19) Beskriv, på en övergripande nivå, vad "the registry" är i Windows 2000 och vad det används till? (1p)
- 20) Vad händer om en process i UNIX försöker öppna en fil flera gånger samtidigt? (1p)

"Hur mycket vore möjligt om ingen försökte det omöjliga?"
 Arne Hirdman

Tentamen i 2G1504 (fd 2G1113) Operativsystem

torsdag 2004-08-19 kl 0900-1300 Lösningsförslag

- 1) Till en process finns knutet ett process control block (PCB), Informationen i PCB kan generellt delas in i tre kategorier - vilka? (1.5p)

- i) Processhantering, t.ex register, programräknare, stackpekare, PID
- ii) Minneshantering, sidtabeller
- iii) Filhantering, rootdirectory, fildescriptorer, användare och grupp identiteter

- 2) Rita en schematisk bild av adressöversättningen i ett system med sidindelning och TLB och en-nivåsidtabell. (1p)

Se föreläsningssanteckningarna

- 3) I ett system finns både korta jobb som i medeltal exekverar 0.1 sek innan de blockerar på I/O eller är klara, och långa jobb som i medeltal exekverar 10sek-10min. Diskutera för- och nackdelar med att använda schemalägningsstrategierna FCFS, SJF respektive round-robin för systemet (3p)

FirstComeFirstServe: enkel att implementera men kan ge dålig responstid för korta/interaktiva jobb, ger inte svältning

ShortestJobFirst: Kräver att jobben kan anmäla hur länge de skall exekvera - vilket kan vara svårt att veta i förväg. Om många korta jobb kommer riskeras svältning av längre jobb. Ger antagligen bra responstider för korta jobb.

Round-robin: Om systemet inte har time-slicening (tidsdelning) så kommer det att fungera pss som FCFS

- 4) I schemalägningsystem talas ofta om måttet "throughput". Vad innebär det? När är det meningsfullt? (1p)

Antal jobb som slutförs per tidsenhet. Meningsfullt i batch-system, dvs system där man inte har interaktivitet.

- 5) Kan en vanlig användare påverka schemaläggningen i LINUX/UNIX? Hur? (1p)

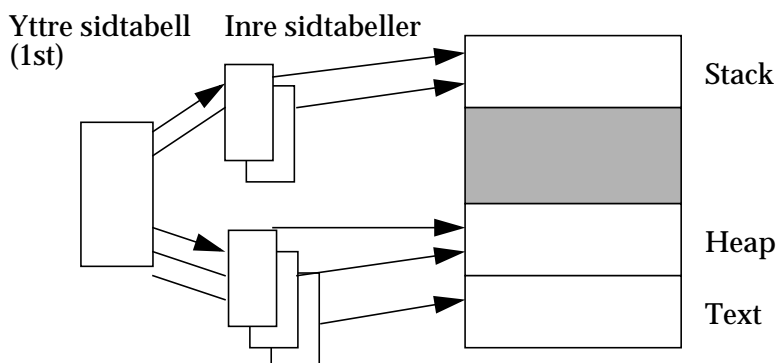
Man kan sänka prioritet för en process genom att ange ett "NICE"-värde som är positivt

- 6) I vilken del av en modern datorarkitektur implementeras normalt adressöversättningen i det fall man har virtuelltminne baserat på sidindelning? (1p)

Normalt sett i MMU:n

- 7) Hur bör man dela upp en virtuell adress för att minimera utrymmet som krävs för en två-nivåers sidtabell givet en virtuell adressrymd om 2^{32} , en fysisk adressrymd om 1GB, sidstorlek på 4KB, lika många bitar (β) används för en post (entry) i den yttre som inre sidtabellen, och att en genomsnittlig process använder 16MB för stack, 8MB data och 4MB för text? (2p)

Sidstorleken $4KB = 2^{12}$ B ger att vi har 20 bitar att använda till att adressera yttre resp. inre sidtabell. Eftersom en post i såväl yttre som inre sidtabell upptar lika många bitar (β) kan vi räkna på storleken i "enheten" antal poster. Vidare kan man konstatera att stacken ($16MB = 2^{24}B$, dvs 2^{12} sidor a 4KB) ligger i en ände av den virtuella adressrymden och heap och text ($8MB + 8MB = 16MB$) intill varandra i en annan ände av adressrymden. En schematisk bild av sidtabeller och adressrymd blir då:



Minnesutrymmet som krävs för sidtabellerna är en yttre sidtabell + det antal inre sidtabeller som går åt för att adressera stack respektive heap+text. Vi kan tabellera minnesutrymmet för olika uppdelningar av den virtuella adressen:

Table 1:

yttre	inre	#inre för stack	#inre för heap+text	Storlek på sidtabeller i antal poster
2^7	2^{13}	1	1	$2^7 + 2*2^{13} = 16.125K$
2^8	2^{12}	1	1	$2^8 + 2*2^{12} = 8.25K$
2^9	2^{11}	2	2	$2^9 + 4*2^{11} = 8.5K$
2^{10}	2^{10}	4	4	$2^{10} + 8*2^{10} = 9K$
2^{11}	2^9	8	8	$2^{11} + 16*2^9 = 10K$

Optimalt är alltså att använda 8 bitar till att adressera den yttre sidtabellen och 12 bitar för de inre sidtabellerna.

- 8) *Beskriv begreppen dynamisk laddning respektive dynamisk länkning!* (2p)

Dynamisk laddning: modulen laddas under run-time vid behov. Dynamisk länkning: modulen laddas/länkas in vid behov men om någon annan process redan har modulen i minne delas denna.

- 9) *Hur brukar man säkerställa att en användare inte kan exekvera en fil med t.ex data eller text även om den har rättigheter i filsystemet (protection) som medger exekvering?* (1p)

Man har ett speciellt internt format på exekverbara filer. Ofta innehåller dette ett "magic number" på en specifik position. Om en fil man försöker exekvera inte har detta interna format, och speciellt rätt magic number, avbryts laddningen och därmed exekveringen.

- 10) i) *I filsystemssammanhang används ofta en "block cache". Vad är det?* (1p)

Cache i primärminnet för block från filsystemet - för att minska disktrafiken och därmed öka prestandan.

- ii) *Varför vill man inte ha ren LRU som utbytesstrategi för block cachern?* (1p)

För att ofta använda block ofta är viktiga för filsystemet, och filsystemskonsistensen. T.ex skulle block innehållande i-noder inte skrivas tillbaka tillräckligt ofta om LRU används för att man skall ha tillräckligt stor chans att klara filsystemskonistens om datorn går ned.

- 11) *Kalle hävdar att context switch tiden att byta mellan två kärntrådar är betydligt mindre än att byta mellan två processer - har han rätt eller fel? Förklara någorlunda utförligt varför?* (2p)

Ja Kalle har rätt - om man byter mellan processer är det ganska mycket data som byts, t.ex sidtabeller, tabeller med fildescriptorer etc. Dessa delas av trådar i en och samma process och vid trådbyte behövs man i princip bara byta stack och programpekare och registerinnehåll.

- 12) *Antag att vi har ett system med sidstorlek som rymmer 500 heltal av typen `int`. Systemet har ett primärminne med 3 ramar. Nedanstående programs textarea rymmer i en ram. Hur många sidfel genereras, givet att matriser lagras radvis och sidutbytesalgoritmen är LRU, när respektive program i och ii exekveras om vi har ren*

Program i	Program ii
<pre>void main(int argc, char **argv) { int arr[50][100]; int i, j; for(i=0; i<50; i++) for(j=0; j<100; j++) arr[i][j] = 0; }</pre>	<pre>void main(int argc, char **argv) { int arr[50][100]; int i, j; for(i=0; i<100; i++) for(j=0; j<50; j++) arr[j][i] = 0; }</pre>

Texten ligger på en sida, variablerna i,j ligger på en sida med de första 498 heltalen i vektorn.

1 sidfel för att få in texten

1 sidfel för att skapa stacken

1 sidfel för att få in i,j och delar av matrisen arr.

För program i som går igenom matrisen radvis får man sedan ytterligare 10 sidfel. För program ii får man istället 10 sidfel per inre loop - dvs 1000 sidfel.

Svar i) 13 ii) 1003 sidfel

- 13) Kalle skall designa ett lastbalanseringsystem för ett multiprocessor system. Han brottas med en grundläggande designfråga där han har två val:

i) En CPU som är överlastad försöker lämna jobb till sina grannar för att minska sin last

ii) En CPU som är underlastad frågar sina grannar om jobb att utföra

Vilken strategi bör Kalle välja?

(1p)

Strategi i leder till mer last för en redan överlastad CPU och om alla är överlastade slutar det med att ingen utför något vettigt arbete alls. Strategi ii lider inte av dessa problem och bör alltså väljas.

- 14) Beskriv stegen i ett remote procedure call!

(1.5p)

Se föreläsningssanteckningar om distribuerade system

- 15) Bankir algoritmen kan användas för att undvika deadlock genom att hålla ett system i "säkert" tillstånd. Givet följande system, vad är det minsta värdet på X som gör att systemet är i ett säkert tillstånd?

(1p)

Process	Allokerat			Max behov			Tillgängligt		
A	1	2	3	6	9	4	2	X	2
B	0	4	2	3	7	5			
C	1	1	4	3	2	4			
D	2	2	3	5	8	9			

Om $X = 1$ kan C exekvera klart och lämna tillbaka de resurser den håller, dock finns ingen process som kan garanteras bli klar med mindre än att 3 enheter av denna typen blir lediga. $X=2$ ger en möjlig väg att avsluta alla processer så att de får sina maxbehov uppfyllda (C, B, D, A)

- 16) I säkerhets sammanhang nämns ofta följande begrepp/namn:

i) CERT

Computer Emergency Response Team - en organisation bildad efter Morris Internet Worm incidenten dit man kan anmäla säkerhetshål/problem och från vilken man kan få information om hur dessa bör åtgärdas. Ett

forum för systemadministratörer.

ii) *Orange book*

USA's försvars krav på säkerhetsåtgärder i bland annat OS. Viktigt att uppfylla dessa krav för att få sälja till statliga myndigheter i USA och många andra organisationer använder den också som standard för krav på datasäkerhet vid upphandlingar.

Beskriv dem kortfattat!

(2p)

17) i) *Beskriv det vanligaste sättet att skapa digitala signaturer.*

(2p)

Man skapar en digital signatur genom att först skapa ett hash-värde H för dokumentet man vill signera. Hash funktionen skall vara svår (omöjlig) att reversera. Hash värdet (de) krypteras sedan med den privata nyckeln i ett public-key kryptosystem. Mottagaren beräknar motsvarande hash-värde H för dokumentet utan signatur och krypterar sedan signaturen med den publica nyckeln - detta värde skall då vara identiskt med H.

ii) *Vilket skydd ger det användaren av en applet om den är försedd med en digital signatur?*

(1p)

Det ger inget skydd i sig - det talar bara med stor sannolikhet om att appleten inte förvanskats sedan den som signerade den satte sin signatur på den.

18) *När är ett periodiskt realtidsystem schemalägningsbart?*

(1p)

C = maximal beräkningstid för ett jobb, P = jobbet's period. Schemalägningsbart om summan av C/P för samtliga jobb är mindre än eller lika med ett. dvs lasten ≤ 1

19) *Beskriv, på en övergripande nivå, vad "the registry" är i Windows 2000 och vad det används till?*

(1p)

Det är en databas med hård och mjukvaruinställningar för WINDOWS, utan den upphör systemet att fungera.

20) *Vad händer om en process i UNIX försöker öppna en fil flera gånger samtidigt?*

(1p)

Det fungerar utmärkt - man får tillbaka två olika fildescriptorer som i sin tur pekar ut samma strukturer i tabellen med öppna filer i OS:et.

"Hur mycket vore möjligt om ingen försökte det omöjliga?"
Arne Hirdman