
Tentamen i TDIU11 Operativsystem

Datum 2024-08-27

Examinator

Tid 14–18

Filip Strömbäck (filip.stromback@liu.se)

Institution IDA

Administratör

Erika Larsson

Kurskod TDIU11

Jourhavande lärare

Provkod TEN1

Filip Strömbäck (013-28 27 52)

Tillåtna hjälpmedel

Lexikon

Instruktioner

- Inkludera motiveringar, förklaringar, resonemang, och beräkningar. Enbart rätt svar ger inte (full) poäng.
- Var precis i dina påståenden. Svävande eller tvetydiga svar ger ej full poäng.
- Skriv ner eventuella antaganden du gör när du besvarar frågorna.
- Skriv tydligt. Oläsliga svar ger ej poäng.
- Lös endast en numrerad uppgift per inlämnat blad.
- Kom ihåg att fylla i tentaomslaget. Numrera inlämnade sidor och skriv AID-nummer på samtliga inlämnade blad.
- Tentamen har 6 uppgifter på 4 sidor (inklusive denna sida).
- Tentamen är på 40 poäng och betygsätts U, 3, 4, 5 (prel. gränser: 20p, 30p, 35p).
- Lycka till!

Tvåpotenser

$2^0 =$	1	$2^5 =$	32	2^{10} bytes =	1 KiB
$2^1 =$	2	$2^6 =$	64	2^{20} bytes =	1 MiB
$2^2 =$	4	$2^7 =$	128	2^{30} bytes =	1 GiB
$2^3 =$	8	$2^8 =$	256	2^{40} bytes =	1 TiB
$2^4 =$	16	$2^9 =$	512	2^{50} bytes =	1 PiB

Schemaläggning

1. Ett system med en CPU kör tre trådar, T1–T3. I början av tick 0 ser systemet ut som i tabellen nedan. Systemets ready-kö innehåller T1 och T2 (i den ordningen).

Tråd	Exekeveringstid	Status
T1	9	Redo
T2	4	Redo
T3	5	Väntar på disk

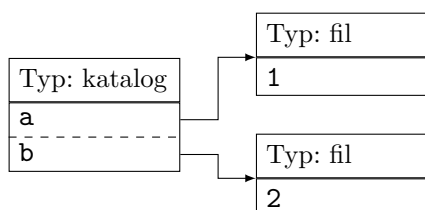
När systemet körs inträffar följande händelser. Antag att händelserna sker precis innan den tidpunkt som anges i tabellen.

Tick	Händelse
3	Data från disk anländer
5	Den aktiva tråden ber att få sova till början av tick 10
6	Den aktiva tråden ber om data från nätverket och får vänta
13	Data från nätverket anländer

- (a) Systemets schemaläggare använder *round-robin* med ett tidskvantum på 2 tick. Trådar som blir redo att köras läggs sist i ready-kön. De avbryter inte körande tråd. Rita ett Gantt-schema över hur trådarna ovan blir schemalagda. Ange även innehållet i systemets ready-kö varje gång den ändras. [3p]
- (b) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (a)? [1p]
- (c) Hur stor andel av tiden i intervallet 0–19 (inklusive både tick 0 och tick 19) används CPU enligt schemaläggningen i (a)? Dvs. vad är *CPU utilization* i intervallet 0–19? [1p]
- (d) Hur schemaläggs trådarna om *shortest job first* (utan preemption) används i stället? Rita ett Gantt-schema och ange innehållet i systemets ready-kö varje gång den ändras. [3p]
- (e) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (d)? [1p]

Filsystem

2. Ett filsystem i ett Linux-system innehåller två filer, **a** och **b**. Båda ligger i filsystemets rotkatalog. Filen **a** innehåller texten 1 och filen **b** innehåller texten 2. Filsystemet innehåller alltså de inoder som visas i bilden nedanför till vänster. Ett program i systemet utför sedan operationerna nedanför till höger: [2p]



- 1: Skapa en hård länk **h** till filen **b**
- 2: Skapa en symbolisk länk **s** till filen **h**
- 3: Ta bort **a**
- 4: Ta bort **b**
- 5: Byt namn på **s** till **a**
- 6: Byt namn på **h** till **b**

- (a) Rita hur katalogen ser ut efter steg 2 och efter steg 6 ovan. Rita katalogen på ett liknande sätt som i bilden ovanför. [2p]
- (b) Vad händer om ett program försöker öppna filerna **a** och **b** efter steg 6? Om filen går att öppna, ange filens innehåll. Om inte, förklara varför det inte går och vad som händer i stället. [2p]
- (c) Hur vet filsystemet när en inod kan markeras som ledig på disk? I vilket eller vilka steg ovan sker det? [1p]

3. Du har en mekanisk hårddisk som är 4 TiB stor (2^{42} bytes). Disken är uppdelad i fysiska block som är 4 KiB stora. Operativsystemet kan välja att använda en logisk blockstorlek som är större än den fysiska blockstorleken, så länge den är en multipel av den fysiska blockstorleken.
- (a) Du vill formatera disken med filsystemet FAT32. Hur stora ska de logiska blocken vara för att FAT-tabellen ska vara 64 MiB stor? [2p]
 - (b) En stor blockstorlek (exempelvis den du kom fram till i (a)) kan leda till att disken nyttjas ineffektivt. Hur illa det blir beror på vad som lagras på disken. Ge två exempel på innehåll på disken (typ av filer, antal filer) där... [2p]
 - 1. ...majoriteten av diskens utrymme **inte kan** användas.
 - 2. ...majoriteten av diskens utrymme **kan** användas.
 - (c) Antag att filsystemet som används på disken använder tre nivåers indexerad allokering, 32-bitars blockindex, och 4 KiB block. Hur stora filer kan maximalt lagras i filsystemet? (Dvs. vad är den maximala filstorleken?) [2p]
 - (d) Indexerad adressering gör det möjligt att använda mindre blockstorlek jämfört med FAT. Använd det innehåll du föreslog i fall 1 i fråga (b) för att jämföra hur mycket utrymme som inte kan användas för FAT jämfört med tre nivåers indexerad allokering. [2p]
- Tips:** Fundera på hur mycket utrymme krävs för att lagra *en* av filerna i ditt exempel med FAT respektive tre nivåers indexerad allokering.

Säkerhet

4. För att hålla systemet säkert måste operativsystemet kunna begränsa vad olika program får göra. För att ådstakomma detta tar operativsystemet hjälp av olika mekanismer som finns i hårdvaran. Nedan beskrivs tre program som alla försöker kringgå skyddet i operativsystemet. För varje program nedan, beskriv:
- 1. Vilka säkerhetsrelaterade problem kan programmets beteende leda till om det inte förhindras av operativsystemet? Ge ett konkret exempel på en situation där programmet kan förstöra för användaren och/eller andra program i systemet.
 - 2. Vilken eller vilka mekanismer från hårdvaran använder operativsystemet för att förhindra programmet?

De tre programmen är som följer:

- (a) Programmet konstruerar först en pekare till regionen av minnet som används av kernel. Programmet skriver sedan till minnet som pekaren refererar till. [2p]
- (b) Programmet använder I/O-instruktioner för att kommunicera med en av diskarna i systemet, och ber disken att läsa eller skriva ett eller flera logiska block. [2p]
- (c) Programmet spenderar all sin tid i en loop som endast involverar beräkningar. Eftersom inga systemanrop körs, så anropas aldrig kod i kernel. [2p]

Minne

5. Du har en processor som producerar 50 bitars logiska adresser. Du vill skapa en modul som med hjälp av paging i 3 nivåer översätter dessa till 40 bitars fysiska adresser.
- (a) Hur stor måste varje page vara för att kunna hantera en logisk adressrymd som är 50 bitar stor? Självklart vill du välja en page-storlek som är så liten som möjligt. Motivera ditt svar, och ange åtminstone också följande information: [2p]
- Vad behöver en rad i en page-tabell innehålla? Hur stor blir varje rad?
 - Hur många rader får plats i en page-tabell, och hur stor blir page-tabellen?
- (b) En process har allokerat 1 GiB virtuellt minne i systemet du designade i (a). Hur mycket plats krävs för att lagra de page-tabeller som krävs för detta? [2p]
6. Ett system med 32-bitars logiska adresser använder virtuellt minne och paging (varje page är 32 KiB). Systemet kan därför lagra pages som inte har använts på ett tag på disk. Program kan därför använda mer minne än vad som får plats i RAM. Systemet har 1 MiB RAM installerat och har tillgång till en stor (> 16 GiB) mekanisk disk.

Följande program körs i systemet ovan:

```
1 int data[1024 * 1024]; // 2^20 element, 4 bytes/element
2
3 int main(void) {
4     // ...
5     for (int i = 1; i < 1024 * 1024; i++) {
6         data[0] = data[0] + data[i];
7     }
8     return data[0];
9 }
```

Antag att bara arrayen `data` behöver lagras i RAM. Antag också att RAM är tomt när loopen i programmet börjar köras (dvs. alla pages lagras då på disk).

- (a) Hur många frames får plats i RAM? [1p]
- (b) Hur många pages krävs för att lagra arrayen `data`? [1p]
- (c) Hur många page-faults sker om loopen i programmet körs på ett system som använder **LRU** för att bestämma vilken page som ska ersättas när RAM är fullt? Motivera ditt svar. [2p]
- (d) Hur många page-faults sker om loopen i programmet körs på ett system som använder **FIFO** för att bestämma vilken page som ska ersättas när RAM är fullt? Motivera ditt svar. [2p]