
Tentamen i TDIU11 Operativsystem

Datum 2025-03-28

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, T2 och T3 (i den ordningen).

Tråd	Exekeveringstid
T1	6
T2	4
T3	5

Under körning så anropar trådarna systemanrop för att be operativsystemet göra saker enligt nedan.

Tråd	Körtid	Händelse
T1	4	Tråden försöker läsa data från tangentbordet och får sova. Data kommer precis innan tick 15, och tråden väcks då.
T3	1	Tråden ber att få sova till precis innan tick 14.

Kolumnen *Körtid* motsvarar antal tick som processen har körts på CPU. T3 ber alltså att få sova precis innan den har körts i 1 tick, och T1 precis innan den har körts i 4 tick.

- (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 schemaläggs. Se till att det framgår vilka trådar som finns i systemets ready-kö vid varje tidpunkt. [3p]
- (b) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (a)? [1p]
- (c) Hur schemaläggs trådarna om *shortest remaining time first* används i stället? Rita ett Gantt-schema över hur trådarna schemaläggs. Se till att det framgår vilka trådar som finns i systemets ready-kö vid varje tidpunkt. [3p]
- (d) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (c)? [1p]
- (e) Hur stor andel av tiden i de 20 första ticken används CPU enligt schemaläggningen i (a) respektive (c) ovan? Dvs. vad är *CPU utilization* i (a) respektive (c)? [1p]

Minne

2. Du har fått i uppgift att bygga en paging-enhet för en CPU med 62 bitars logiska adresser. Resten av systemet kräver att element i page-tabeller är 4 bytes stora, och att det finns plats för en valid-bit och en bit för att indikera att motsvarande page bara ska vara tillgänglig i kernel-läge. Utöver dessa krav ska pages vara maximalt 32 KiB stora för att minimera intern fragmentering, och alla frames som används för att lagra page-tabeller ska kunna nyttjas fullt ut (dvs. ingen intern fragmentering i pagetabellerna).
- (a) Hur många nivåer av paging och vilken pagestorlek ska du välja för att uppfylla kraven ovan? Om flera alternativ är möjliga, välj det med störst pagestorlek. [3p]
 - (b) Vad är den största möjliga storleken för fysiska adresser givet ditt svar i (a)? [1p]
 - (c) Vad är den genomsnittliga accesstiden för minne inklusive overhead från ditt svar i (a)? Antag att varje access till DRAM tar 10 ns, och att det finns en TLB som kan användas i 90% av fallen. [2p]

3. En processor designad för små inbyggda system implementerar virtuellt minne med hjälp av paging. Processorn tillhandahåller 12 bitar logiska adresser. Dessa översätts till 16 bitar fysiska adresser med hjälp av en nivå av paging. Varje page är 256 bytes stor.

På systemet körs för närvarande två processer (process 1 och process 2). Deras page-tabeller ser ut som i tabellerna nedan. Varje rad lagras som 2 bytes, dvs. 16 bitar.

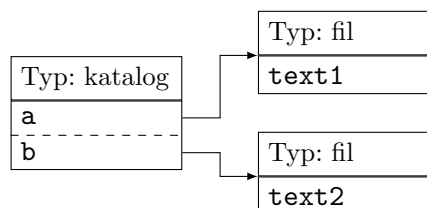
Rad	Process 1		Rad	Process 2	
	Frame	Valid		Frame	Valid
0x0	0x8A	0	0x0	0xD3	1
0x1	0x99	1	0x1	0xAF	0
0x2	0xBD	1	0x2	0xF6	1
0x3	0xD3	1	0x3	0xB7	1
0x4	0x05	1	0x4	0xEE	1
0x5	0x9B	1	0x5	0xCF	1
0x6	0xB0	0	0x6	0x89	1
0x7	0xBC	1	0x7	0x73	1
0x8	0xF4	1	0x8	0x93	1
0x9	0x6B	1	0x9	0x51	1
0xA	0xC7	0	0xA	0xB7	1
0xB	0x59	1	0xB	0x66	1
0xC	0x86	1	0xC	0x74	0
0xD	0x3D	1	0xD	0x71	0
0xE	0xA8	1	0xE	0xAF	0
0xF	0x93	0	0xF	0xF9	1

- (a) Process 1 och process 2 läser adress 0xA31. Hur översätts den virtuella adressen till en fysisk adress? Vad blir resultatet av översättningen i båda fallen? [2p]
- (b) Delar process 1 och process 2 minne? Om så är fallet, ange numret på de frames som är delade. För varje delad frame, ge också två logiska adresser, en för process 1 och en för process 2, som refererar till samma fysiska adress. [2p]
- (c) Finns det flera logiska adresser som refererar till samma fysiska adress i process 2? Om ja, ange också två olika logiska adresser som refererar till samma fysiska adress, och även vilken fysisk adress dessa motsvarar. [2p]

Filsystem

4. Du har en mekanisk hårddisk som är 4 TiB stor, och uppdelad i 512 bytes fysiska block. Operativsystemet kan använda större logiska block genom att slå ihop fysiska block.
- (a) Vilken är den minsta logiska blockstorleken som gör att alla logiska block kan indexeras med ett 32-bitars heltal? [2p]
- (b) Vad är den maximala filstorleken om 2 nivåers indexerad allokering används? [2p]
- (c) Antag att disken har 10 ms accesstid oavsett blockstorlek. Hur lång tid tar det att öppna en fil och läsa det 100:e blocket om (i) länkad allokering används, och om (ii) 2 nivåers indexerad allokering används? [2p]
- (d) Antag att disken schemaläggs med hjälp av SSTF (Shortest Seek Time First). De processer som körs i systemet vill läsa följande block: 600, 880, 160, 650, 360, 670. Diskens läshuvudet befinner sig nu på position 620. I vilken ordning schemaläggs läsningarna och vad blir den totala sträckan som läshuvudet behöver röra sig? [2p]

5. Rotkatalogen i ett filsystem på ett Linux-system innehåller två filer, **a** och **b**. Filerna innehåller **text1** och **text2** enligt figuren nedanför till vänster. Ett program i systemet utför sedan operationerna nedanför till höger:



- 1: Skapa en symbolisk länk **s** till **a**
- 2: Öppna **s** som fildeskriptor 1
- 3: Ta bort **a**
- 4: Byt namn på **b** till **a**
- 5: Skriv **newcontents** till fildeskriptor 1
- 6: Stäng fildeskriptor 1

- (a) Rita hur katalogen ser ut efter steg 2 och efter steg 6 ovan. Använd en notation som liknar figuren ovan. Inkludera också vilken fil som fildeskriptor 1 refererar till i steg 2. [2p]
- (b) Vad händer om ett program efter steg 6 försöker öppna **s**? Om det går, ange innehåll i filen som öppnades. Om inte, förklara vad som går fel. [2p]
- (c) I vilket eller vilka steg ovan markeras en inod som ledig? [1p]

Säkerhet

6. Du har blivit ombedd att bygga ett system för att hantera inlämning av datorlaborationer. Systemet ska lagra data i katalogen **/handins**, som finns på ett filsystem som delas mellan lärare och studenter. Utöver detta finns två program: **submit** och **grade**. Programmet **submit** körs av en student för att lämna in en labbuppgift. Programmet kopierar då uppgiften till filen **/handins/<student-id>** och sätter lämpliga filrättigheter. Programmet **grade** körs av en lärare för att hämta alla labbar. Programmet letar då i **/handins** och kopierar alla labbar till lärarens konto. Du kan anta att alla studenter har sina egna användarkonton, och att systemet bara används i en kurs.

Systemet ska uppfylla följande krav:

- Studenter ska *inte* kunna läsa andra studenter inlämningar. Det är dock okej om en student kan läsa eller uppdatera sin egen inlämning.
- Programmet **submit** ska inte behöva vara setuid eller ha särskilda capabilities. IT-avdelningen anser att risken med detta är för stor.

- (a) Beskriv hur du kan använda lämpliga mekanismer som finns i Linux-system (ex.vis filrättigheter, ACL, capabilities, setuid) för att uppfylla kraven i systemet. Tänk specifikt på: [4p]
- Behöver speciella grupper skapas av systemadministratörerna? Om ja, vilka grupper behövs, och vilken/vilka användare ska vara med i grupperna?
 - Vilka rättigheter ska katalogen **/handins** ha?
 - Vilka filrättigheter ska programmen **submit** och **grade** ha?
 - Vilka filrättigheter ska **submit** sätta på labbar som lämnas in (dvs. på filen **/handins/<student-id>**)?

- (b) Vad händer om en student lämnar in en labb genom att själv kopiera den till **/handins**? Kan läraren hämta labben med **grade** i det fallet? [1p]

Notera: Din lösning i (a) behöver *inte* lösa detta problem.

- (c) Vad händer om en student försöker ta bort alla filer i **/handins** (ex.vis med kommandot **rm /handins/***)? [1p]

Notera: Din lösning i (a) behöver *inte* lösa detta problem.