
Tentamen i TDIU11 Operativsystem

Datum 2025-08-26

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 7 uppgifter på 5 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. Vid tid $T = 0$ ser systemet ut som i tabellen nedan. Kolumnen *uppskattad exekeveringstid* anger hur många tick som systemet uppskattar att tråden kommer köra när den hamnar i ready-kön (schemaläggaren kan inte se in i framtiden och kan därför inte veta hur länge varje tråd faktiskt kommer att köra).

Tråd	Tillstånd	Uppskattad exekeveringstid
T1	I väntekö för disk	4
T2	I väntekö för nätverk	2
T3	I ready-kö	8

Följande saker händer i systemet under körning. Antag att händelserna sker precis innan det tick som anges i *tid*-kolumnen (dvs. data från disk anländer precis innan tick 3).

Tick	Händelse
3	Data från disk anländer
4	Data från nätverk anländer

När trådarna får exekevera på CPU:n gör de följande saker. Kolumnen *exekeveringstid* anger hur många tick som varje tråd har kört på CPU:n innan de gör vad som står i kolumnen *händelse*. Exempelvis kommer T2 att terminera precis innan den har exekeverat i två tick.

Tråd	Exekeveringstid	Händelse
T1	3	Tråden skriver till disk och får vänta
T2	2	Tråden terminerar
T3	9	Tråden läser från tangentbordet och får vänta

- (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? Notera att schemaläggaren inte kan se in i framtiden och måste därför förlita sig på den uppskattade exekeveringstiden, även om den inte alltid är helt korrekt. 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]

Minne

2. Du har fått i uppgift att bygga en paging-enhet till en CPU med 38 bitar logiska adresser. Systemet ska kunna använda 512 GiB fysiskt minne. Systemet kräver att element i alla pagetabeller är 4 byte stora, och att varje element innehåller minst 6 bitar för metadata (ex.vis valid-bit och dylikt).

(a) Vad är den minsta möjliga pagestorleken för att CPU:n ska kunna använda allt fysiskt minne? [2p]

(b) Hur många nivåer av paging ska användas i systemet? Din första prioritet är att undvika intern fragmentering i frames som lagrar page-tabeller (dvs. varje pagetabell ska vara full). Om det finns flera möjliga alternativ som inte har någon fragmentering ska du välja det som ger så små pages som möjligt (men som respekterar begränsningen i (a)). Visa ditt resonemang, och varför ditt val minimerar pagestorleken. [3p]

(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 tillhandahåller virtuellt minne med hjälp av 2 nivåer paging. Processorn arbetar med 6 bitar logiska adresser och 8 bitar fysiska adresser. De logiska är uppdelade i $2 + 2 + 2$ bitar. Varje element i en pagetabell är 1 byte stor, och varje page är 4 bytes stor.

Nedan visas de pagetabeller som finns i systemet. Rubriken på varje tabell indikerar vilken frame den lagras i. Alla tal i uppgiften är skrivna i binär notation. När process 1 körs så är registret som pekar på första nivåns pagetabell satt till 000001.

Lagras i frame 000001		
Rad	Frame	Valid
00	101111	1
01	001100	0
10	010010	1
11	001100	1

Lagras i frame 001100		
Rad	Frame	Valid
00	000000	1
01	001010	1
10	110000	1
11	001100	0

Lagras i frame 010010		
Rad	Frame	Valid
00	111111	1
01	110011	1
10	010010	0
11	000000	0

Lagras i frame 101111		
Rad	Frame	Valid
00	100011	1
01	001100	0
10	111111	1
11	101100	0

(a) Process 1 läser de logiska adresserna nedan. Hur sker översättningen, och vad blir resultatet av översättningen i vart och ett av fallen? [4p]

Logiska adresser: 000000, 101010, 010011

(b) Finns det flera logiska adresser som refererar till samma fysiska adress i process 1? Om ja, ange också två olika logiska adresser som refererar till samma fysiska adress, samt vilken fysisk adress detta motsvarar. [2p]

Filsystem

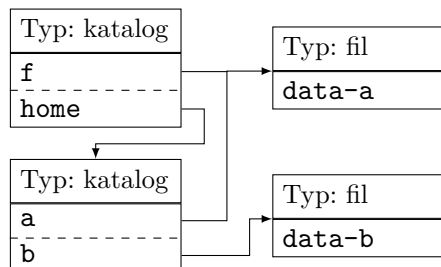
4. Du har en 4 TiB stor mekanisk hårddisk som är uppdelad i 1 KiB fysiska block. Operativsystemet kan använda större logiska block genom att slå ihop 2^n fysiska block (där n är ett heltal och $n \geq 0$). Söktiden är 10 ms. Att läsa ett block tar sedan 10 μ s.
- (a) Vilken är den minsta möjliga logiska blockstorleken för att göra det möjligt att indexera alla logiska block med en 32-bitars pekare? [1p]
- (b) Du vill formatera disken med FAT32. Vilken är den minsta möjliga blockstorleken för att FAT ska vara mindre än 1 GiB? [2p]
- (c) Vad blir den maximala filstorleken om du använder ett filsystem som använder indexerad allokering i 2 nivåer och 32-bitars blockpekare? Filsystemet använder den logiska blockstorleken du kom fram till i (a). [2p]
- (d) Antag att du använder ett filsystem som använder den logiska blockstorleken du kom fram till i (a). Hur lång tid tar det att läsa det sista blocket i en fil som är 4000 KiB stor om filsystemet använder:
- länkad allokering
 - indexerad allokering (som i (c))

Antag att varje läsning av ett nytt *logiskt* block kräver en sökning. Om ett logiskt block består av flera fysiska block kan de sedan läsas direkt efter varandra utan en sökning. Att läsa ett 2 KiB logiskt block tar alltså $10\text{ms} + 2 \cdot 10\mu\text{s}$.

5. Ett filsystem på ett Linux-system ser ut som bilden nedan. Katalogen högst upp till vänster motsvarar filsystemets rot (dvs. sökvägen /). Som framgår i bilden så innehåller rotkatalogen en fil, **f**, och en katalog, **home**. Katalogen **home** innehåller två filer, **a** och **b**. Notera att filen **/home/a** är densamma som **/f**.

Ett program i systemet utför operationerna nedanför till höger:

Filsystemets rot



- 1: Öppna filen **/f** som fildeskriptor 1
- 2: Ta bort filen **/f**
- 3: Skapa en symbolisk länk **/e** till **/home/a**
- 4: Skapa en hård länk **/f** till **/home/b**
- 5: Ta bort filen **/home/a**
- 6: Ta bort filen **/home/b**
- 7: Skriv **ny-data** till fildeskriptor 1
- 8: Stäng fildeskriptor 1

- (a) Rita hur filsystemet ser ut efter steg 4 och steg 8 ovan. Använd en notation som liknar figuren ovan. Indikera också vad fildeskriptor 1 refererar till i steg 4. [2p]
- (b) Vad händer om ett program efter steg 8 ovan försöker öppna filerna **/e** och **/f**? För båda filerna ange om det går att öppna filen. Om det går, ange också innehåll i filen. Om inte, förklara vad som händer i stället. [2p]
- (c) Vad händer om programmet försöker skapa en hård länk **/dir** som leder till **/home**? [1p]

Säkerhet

6. På ett universitet finns ett filsystem som går att komma åt från alla Linux-datorer på campus (både studenter och lärare har tillgång till det). Lärarna på universitetet har länge önskat att det finns *kurskataloger* för alla kurser som ges som läraren kan använda för att distribuera kursmaterial till studenterna. Du har nu fått i uppdrag att designa systemet på ett lämpligt sätt. Din lösning ska uppfylla följande krav: [5p]

- Det finns en katalog `/courses` som innehåller alla kurskataloger, en för varje kurs. Exempelvis är kurskatalogen för TDIU11 `/courses/TDIU11`. Alla ska kunna se innehållet i `/courses`, men endast IT-avdelningen (dvs. administratörer) ska kunna lägga till eller ta bort kurskataloger.
- Den lärare som är ansvarig för en kurs ska kunna läsa och modifiera innehållet i kurskatalogerna för sina kurser. Exempelvis ska Filip kunna manipulera innehållet i `/courses/TDIU11` och `/courses/TDIU16`.
- Alla som läser en kurs ska kunna läsa (men inte modifiera) innehållet i motsvarande kurskatalog. Exempelvis borde du kunna läsa `/courses/TDIU11` eftersom du läser kursen TDIU11. Det är okej om alla användare kan se vilka filer som finns i alla kataloger, så länge de inte kan läsa innehållet i filerna.
- Läraren för en kurs ska kunna göra en del av filerna i en kurskatalog läsbara för alla användare på universitetet. Exempelvis kanske det finns en fil `/courses/TDIU11/kursinfo` som läraren vill att alla ska kunna läsa.

Beskriv hur du kan använda lämpliga mekanismer som finns i Linux-system (ex.vis filrättigheter, ACL, capabilities, setuid) för att uppfylla kraven ovan. Använd gärna ett konkret exempel för att illustrera din lösning (exempelvis med kurserna TDIU11, TDIU16 och någon ytterligare kurs som hålls av en annan lärare).

- Behöver några grupper skapas? Om ja, vilka grupper behöver skapas, och vilken/vilka användare ska vara med i grupperna?
- Vilka rättigheter ska katalogen `/courses` ha?
- Vilka rättigheter ska de olika kurskatalogerna ha? (dvs. `/courses/<kurskod>?`)
- Vilka rättigheter ska filer i en kurskatalog ha? Både de filer som bara ska kunna läsas av studenter som läser kursen (ex.vis `/courses/TDIU11/homework3`), och de filer som ska kunna läsas av alla (ex.vis `/courses/TDIU11/kursinfo`)?

Notera: Fokusera på hur du använder de valda mekanismerna när du anger rättigheter för filer och kataloger. Använder du filrättigheter vill du exempelvis ange både vilken användare och grupp som äger varje fil och vilken åtkomst som ägare, grupp, och andra har. Använd gärna notation liknande vad `ls -l` producerar. Använder du setuid så vill du ange vilket eller vilka program som har setuid, och vad programmen gör.

7. Du är ansvarig för ett system som lagrar användarnamn och lösenord för inloggning. Systemet lagrar för närvarande lösenorden i en databas krypterade med en nyckel som lagras i filsystemet. [1p]
- (a) Beskriv kort (2–3 meningar) varför detta är problematiskt ifall någon kommer åt all data i systemet. [1p]
- (b) Föreslå ett bättre sätt att lagra lösenorden för att minimera risken att problemet i (a) uppstår. [1p]