

---

# Tentamen i TDIU11 Operativsystem

---

**Datum** 2024-03-23

**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 fyra trådar, T1–T4. I början av tick 0 ser systemet ut som i tabellen nedan. Systemets ready-kö innehåller T1 följt av T4.

Tråd	Exekeveringstid	Status	Prioritet
T1	10	Redo	1
T2	5	Väntar på disk	4
T3	4	Väntar på nätverk	3
T4	2	Redo	5

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
7	Data från nätverk 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. [2p]
- (b) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (a)? [2p]
- (c) Hur schemaläggs trådarna om systemet i stället använder prioritetsbaserad schemaläggning med preemption. Högre siffra indikerar högre prioritet. T4 har alltså högst prioritet i systemet. Ange även innehållet i systemets ready-kö varje gång den ändras. [2p]
- (d) Vad blir väntetiden (*waiting time*) för var och en av trådarna i ditt svar i (c)? [1p]
2. I en av de två schemalägningsalgoritmerna i fråga 1 kan *starvation* uppstå.
- (a) I vilken algoritm kan *starvation* uppstå? Illustrera hur detta kan ske med hjälp av ett exempel likt det i uppgift 1. [2p]
- (b) I vilken algoritm kan *starvation* **inte** uppkomma? Motivera ditt svar. [2p]

## Filsystem

3. Du vill formatera en 4 TiB ( $2^{42}$  bytes) stor SSD med ett lämpligt filsystem. Disken är uppdelad i fysiska block som är 4 KiB stora. Filsystemet använder logiska block som är lika stora som de fysiska blocken, dvs. 4 KiB.
- (a) Hur många block finns på disken? Hur många bytes behövs för att lagra ett blocknummer i filsystemet? [2p]
- (b) Antag att vi använder ett filsystem med en FAT (File Allocation Table). Hur stor blir FAT? [2p]
- (c) Antag att vi använder ett filsystem som använder indexerad allokering, med två nivåer av indexblock. Hur stora filer kan i så fall lagras i filsystemet? [2p]
- (d) Indexerad allokering, likt i (c), kräver förhållandevis mycket utrymme för att lagra indexblock, speciellt för små filer. Att lagra en fil som är 4 KiB stor kräver exempelvis att vi lagrar 2 indexblock som är 4 KiB vardera. Beskriv hur vi kan minska denna overhead för små filer, utan att förlora möjligheten att lagra filer som är åtminstone 10 GiB stora. Beräkna också den maximala filstorleken för din lösning. [4p]

## Minne

4. 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.

Process 1			Process 2		
Rad	Frame	Valid	Rad	Frame	Valid
0x0	0x10	0	0x0	0x58	1
0x1	0x59	1	0x1	0x10	1
0x2	0x12	1	0x2	0x2A	0
0x3	0x03	1	0x3	0x2B	0
0x4	0xE1	1	0x4	0x31	1
0x5	0x81	0	0x5	0x55	1
0x6	0x34	0	0x6	0x56	1
0x7	0x77	1	0x7	0x9B	1
0x8	0xA3	1	0x8	0x00	1
0x9	0x42	1	0x9	0xB3	1
0xA	0x92	0	0xA	0xCA	1
0xB	0x31	1	0xB	0xAD	1
0xC	0x3A	1	0xC	0xD3	0
0xD	0x3F	1	0xD	0xE7	0
0xE	0x70	1	0xE	0x68	0
0xF	0x4E	0	0xF	0x21	1

- (a) Både process 1 och process 2 läser adress 0x831. Vilken/vilka fysiska adresser motsvarar detta? Beskriv även hur en logisk adress översätts till en fysisk adress. [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) Processorn innehåller en TLB för att minska kostnaden av paging. Hur lång tid tar varje minnesåtkomst av en logisk adress (dvs. inklusive uppslagning i page-tabell), givet att 80% av adresserna går att hitta i TLB, och att varje minnesåtkomst till RAM tar 10 ns? Antag att tiden det tar att leta i TLB är försumbar. [2p]
5. Du har fått i uppgift att designa modulen som hanterar paging i en processor. Processorn producerar logiska adresser som är 38 bitar stora. Modulen för paging ska utifrån detta använda 2 nivåer av page-tabeller för att producera 40 bitar fysiska adresser.
- (a) Hur bör paging-systemet se ut för att implementera 2 nivåer paging enligt ovan? Motivera dina val, och ange åtminstone: [4p]
- Hur stor ska en page vara?
  - Vad innehåller en rad i en page-tabell? Hur stor blir raden?
  - Hur många rader får plats i en page-tabell?
- (b) En process använder 1 MiB logiskt minne som är allokerat sekventiellt (dvs. i logiska adresser som ligger efter varandra). Hur mycket minne krävs för att lagra de page-tabeller som behövs för detta? Utgå från den design du kom fram till i (a). [2p]

## Säkerhet

6. Ett universitet har beslutat sig för att digitalisera hanteringen av tentor. I stället för att skicka papperskopior av tentor har universitetet beslutat sig för att använda det centrala filsystemet i stället. Processen för tentahantering ser ut som följer:

- Examinator skapar tentan. Tentan läggs i filsystemet som `/exams/<datum>/<kurskod>.pdf`. Exempelvis `/exams/2024-03-23/TDIU11.pdf` för denna tentan.
- Personal på tentaservice ansvarar sedan för att tentorna blir utskrivna inför respektive tillfälle. För att göra detta behöver personalen kunna öppna alla filer under `/exams/`.

Det är viktigt att ingen obehörig kan komma åt tentorna. Endast personal ska få komma åt `/exams/-`mappen. Tentaservice ska få läsa alla filer under `/exams/`. Examinatorer ska få skapa nya filer, men varje examinator ska bara få läsa filer som de själva har skapat (systemet används ibland för vidareutbildning av lärare).

- (a) Beskriv hur du med hjälp av rättigheter i filsystemet i UNIX kan se till så att kraven ovan uppfylls. Dvs. vilken ägare och grupp ska filerna och katalogerna ha, samt vilka rättigheter ska filerna och katalogerna ha. [3p]

Du kan anta att följande grupper finns i systemet:

- `employee` – Har alla anställda som medlemmar.
- `exam` – Har all personal på tentaservice som medlemmar.

- (b) Efter ett tag inser universitetet att de vill tillåta studenter att komma åt de tentor som har ägt rum. Det vill säga, studenter ska få komma åt alla filer i `/exams/XXXX-YY-ZZ/` om datumet `XXXX-YY-ZZ` inte är dagens datum eller ett datum i framtiden. Exempelvis ska studenter få komma åt filer i mappen `/exams/2024-03-22/`, men inte `/exams/2024-03-25/`. [2p]

Tyvärr finns inte möjligheten att schemalägga körning av program för att ändra filrättigheter. Det behövs alltså en annan lösning för att ge studenterna åtkomst till tentorna. Beskriv hur du kan ådstakomma detta med hjälp av de mekanismer som finns i ett typsiskt UNIX/Linux-system.

- (c) Vilken eller vilka mekanismer från hårdvaran använder operativsystemet för att se till att program inte kan kringgå rättigheterna som finns i filsystemet? Beskriv även kort hur mekanismerna hjälper till att skydda systemet från andra typer av attacker. [2p]