



UNIVERSITETET I BERGEN

KANDIDAT

126

PRØVE

INF113 0 Innføring i operativsystem

Emnekode	INF113
Vurderingsform	Skriftlig eksamen
Starttid	14.02.2022 08:00
Sluttid	14.02.2022 11:00
Sensurfrist	--
PDF opprettet	14.02.2022 17:59

Exam Information

Oppgave	Tittel	Oppgavetype
i	The general information about the exam	Informasjon eller ressurser

Easy

Oppgave	Tittel	Oppgavetype
1	Linux commands	Flervalg (flere svar)
2	Binary semaphore	Sant/usant
3	Permission bits	Flervalg (flere svar)
4	Inconsistency problem	Sant/usant
5	What is it?	Fyll inn tekst
6	Free space management	Flervalg
7	Concurrency problems	Sant/usant
8	Wait and signal	Sant/usant
9	Threads	Sant/usant
10	General architecture	Fyll inn tekst
11	Components	Fyll inn tekst
12	Memory related errors	Sant/usant
13	Stack vs. Heap	Paring
14	RAID	Sant/usant
15	Data storage	Paring

Middle

Oppgave	Tittel	Oppgavetype
16	Swapping policies	Flervalg
17	Race condition	Flervalg
18	Utilization of CPU	Flervalg
19	Scheduling policy	Fyll inn tall
20	Valid and invalid VA	Dra og slipp
21	Sequential workload	Flervalg

Advanced

Oppgave	Tittel	Oppgavetype
22	Address translation	Langsvar
23	Concurrency	Langsvar
24	Shell scripting	Langsvar

1 Linux commands

Which Linux command produces a result related to processes?

Select one or more alternatives:

☒ fg

☐ cp

☐ pwd

☐ ps

☒ kill

☒ top

Maks poeng: 2

2 Binary semaphore

The initial value of a binary semaphore should be 0.

Select one alternative:

☐ False

☒ True

Maks poeng: 1

3 Permission bits

```
drwxr-x--- 1 Markus SportTeam 2048 April 15 2015 football
```

Select the linux command to allow the members of the SportTeam to rename files, create new files, or delete existing files in the football directory.

Select one or more alternatives:

☒ `chmod g+w football`

☐ `chmod 770 football`

☐ `chmod o+x football`

☐ `chmod 760 football`

Maks poeng: 2

4 Inconsistency problem

In the file system, *bitmaps* keep track of which inodes/data blocks in the disk have been used. The *inode* of a file contains all the meta information about a file. For example, who owns this file? what time was this file last accessed?

Assume that we want to extend a file in the disk and we know that beside writing the extended file content to the disk, both the bitmaps and the inode table in the disk should be updated correspondingly. If the disk crashes after the bitmaps are updated but neither the inode table is updated nor the extended file content has been written to the disk, **memory leaks**.

Select one alternative:

☐ True

☒ False

Maks poeng: 1

5 What is it?

A is a piece of code that accesses a shared variable (or more generally, a shared resource) and must not be concurrently executed by more than one thread.

Maks poeng: 2

6 Free space management

Which policy for managing the free space has the advantage of speed in general?

Select one alternative:

☒ first-fit

☐ best-fit

☐ worst-fit

Maks poeng: 1

7 Concurrency problems

No deadlock implies no starvation.

Select one alternative:

☒ False

☐ True

Maks poeng: 1

8 Wait and signal

When using semaphores in C, the function `sem_post()` does not wait for a particular condition to hold like the function `sem_wait()` does.

Select one alternative:

☐ False

☒ True

Maks poeng: 1

9 Threads

Threads of a process share the same address space and thus can access the same data.

Select one alternative:

☐ False

☒ True

Maks poeng: 1

10 General architecture

An is a program that acts as an interface between a user of a computer and the computer resources.

Maks poeng: 2

11 Components

is the heart of the operating system.

Maks poeng: 2

12 Memory related errors

Buffer overflow may lead to running out of memory.

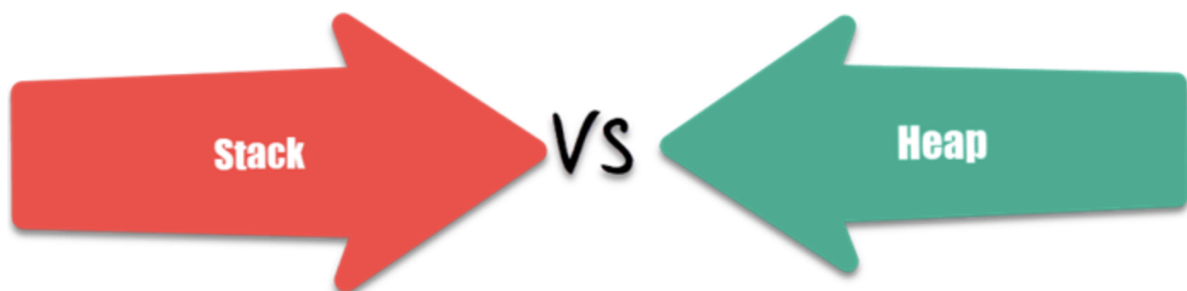
Select one alternative:

☐ False

☒ True

Maks poeng: 1

13 Stack vs. Heap



	stack	heap
This is where the dynamic created variables are stored.	<input type="radio"/>	<input checked="" type="radio"/>
This is where the local variables are stored.	<input type="radio"/>	<input checked="" type="radio"/>
It can be resized.	<input type="radio"/>	<input checked="" type="radio"/>
It grows from high memory to low memory.	<input checked="" type="radio"/>	<input type="radio"/>
Its memory can become fragmented.	<input checked="" type="radio"/>	<input type="radio"/>
This must be used in order to perform recursive function calls.	<input type="radio"/>	<input checked="" type="radio"/>

Maks poeng: 6

14 RAID

RAIDs are designed to detect and recover from certain kinds of disk faults. Below is an example of an RAID-1 system. With N disks, the system tolerates maximum $N/2$ disk failure.

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

Select one alternative:

☒ False

☐ True

Maks poeng: 1

15 **Data storage**

Compare cache, memory and hard disk.

Please match the values:

	cache	memory	hard disk	cache and memory	cache and hard disk	memory and hard disk	all
It has the largest space.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It operates at the highest speed among all three.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It contains pages.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It stores frequently used data.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It contains page frames.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This is where the swap space is located.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Its content of the data is organized by the file system.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is volatile.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Swap-in and swap-out happens here.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A program should be moved here before being executed.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
There is no direct communication between these two.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
It can be virtualized by the operating system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Maks poeng: 12

16 Swapping policies

When the physical memory space is not enough, the OS will swap pages out of memory to the swap space and swap pages into memory from the swap space. Assume a memory can contain at most 3 pages and the memory is empty initially. What is the hit rate when accessing pages 1,0,1,2,3,2,0,3,1,1 one-by-one in a sequence using the First-in First-out (FIFO) swapping policy and Least-Recently-Used (LRU) swapping policy, respectively?

Select one alternative:

- ☐ FIFO (0.5) , LRU (0.4)
- ☐ FIFO (0.4) , LRU (0.5)
- ☒ FIFO (0.3) , LRU (0.4)
- ☐ FIFO (0.5) , LRU (0.3)
- ☐ FIFO (0.3) , LRU (0.5)

Maks poeng: 5

17 Race condition

Assume $x = 1$ and $y = 2$ initially. After the concurrent execution between **thread 1** and **threads 2** finished, how many different combinations of x and y can be ?

thread 1	thread 2
$x = x + y$	$y = y - x$

Select one alternative:

- ☐ 2 possible cases
- ☐ 3 possible cases
- ☐ 5 possible cases
- ☒ 4 possible cases

Maks poeng: 5

18 Utilization of CPU

Assume that it takes one time unit for CPU to issue an I/O instruction, five time units for the I/O device to actually perform the task, and then one more time unit for CPU to complete the I/O instruction. So the total time from issuing an I/O instruction to finish the I/O instruction is $1+5+1=7$ time units, in which 2 time units are spent on the CPU.

For running two processes, i.e., one process executes 1 I/O instruction and the other process executes 4 CPU instructions, compute the percentage of time that CPU is in use in the following two systems, respectively:

1. The system does NOT switch to another process while one is doing I/O, instead waiting until the process is completely finished.
2. The system switches to another process whenever one is WAITING for I/O.

Select one alternative:

☐ 48.28%, 75.12%

☐ 51.55%, 68.23%

☒ 54.55%, 80.45%

☐ 54.55%, 85.71%

Maks poeng: 5

19 Scheduling policy

Compute the average turnaround time when running five jobs of different lengths: 20, 10, 30, 15, 5 with the RR scheduler and a time-slice of 5. Assume 5 jobs arrive in the system at roughly the same time.

What is the average turnaround time of using Round Robin: .

Maks poeng: 5

20 Valid and invalid VA

There are 5 virtual addresses (VA) listed below. Based on the base-and-bounds register information, which VA is valid and which one can cause segmentation violation? Please drag and drop each one of the 5 VAs into the corresponding category. (1 point per correct answer, -1 point per wrong answer, minimum point is 0)

Base-and-Bounds register information:

Base : 0x00000199

Limit : 419

Segmentation Violation	Valid Address
0x000002b4	0x00000119
0x000002f2	0x000000e4
0x00000391	

Maks poeng: 5

21 Sequential workload

Seek time is the time spent to move the drive arm to the correct track. When the correct track is found, it takes some time to find the right sector on the track. This time is called rotational delay.

$$\text{Sequential workload} = \frac{\text{Amount of data}}{\text{Time to access}}$$

Assume for a disk the average seek time is 7 ms, the average rotational delay is 3 ms, and the transfer rate of data is 50 MB/s. What is the sequential workload of such disk for a sequential transfer of size 10 MB?

Select one alternative:

- ☐ 32.12 MB / second
- ☐ 19.48 MB / second
- ☐ 51.78 MB / second
- ☒ 47.62 MB / second

Maks poeng: 5

22 Address translation

Assume the address space size is 16k and the page size 4k. The format of the page table is described as the following: The high-order (left-most) bit is the VALID bit. If the bit is 1, the rest of the entry is the PFN. If the bit is 0, the page is not valid.

Given the page table below, which virtual addresses are valid and which are not?

(1) If it is valid, please write down the corresponding physical address in the hexadecimal format (0x....) and your calculation steps. (8 points)

(2) For the remaining ones, please explain why they are invalid. (4 points)

Page Table (from entry 0 down to the max size)

0x00000000

0x8000000a

0x80000002

0x00000000

Virtual Address:

0x00000e69 --> ?

0x00002efa --> ?

0x00000855 --> ?

0x00002201 --> ?

Fill in your answer here

The two middle values of the page table is valid :))

Maks poeng: 12

23 Concurrency

(1) Explain what the assembly code below does. (8 points)

Hint: Write “What are the variables **a** and **b** used for?”

(2) Say at least two things that the code is good at in a concurrent setting, and why. (4 points)

Note that:

- The initial value of variables **a**, **b**, and **c** are all 0.
- The instruction **fetchadd %ax, a** atomically increments the value of **a** by the value of **ax** and saves the original value of **a** in **ax**.

```
.var a
```

```
.var b
```

```
.var c
```

```
.main
```

```
.top
```

```
.acquire
```

```
mov $1, %ax
```

```
fetchadd %ax, a
```

```
.tryagain
```

```
mov b, %cx
```

```
test %cx, %ax
```

```
jne .tryagain
```

```
mov c, %ax
```

```
add $1, %ax
```

```
mov %ax, c
```

```
mov $1, %ax
```

```
fetchadd %ax, b
```

```
sub $1, %bx
```

```
test $0, %bx
```

```
jgt .top
```

```
halt
```

Fill in your answer here

This code assigns the integer 1 to the %ax, then it adds the value of ax to a (and store the original value of a in ax). Then it assigns b to %cx and compares %cx and %ax. Then if false it goes back to .tryagain if not it continues. Then ax gets the value of c, ax increments with 1 then c gets the value of ax. Then ax gets incremented by 1. b increments with the value of ax and ax gets the original value of b. bx decreases by 1 integer. Then it compares the value of bx and 0 if false the code goes to .top if not the code halts.

Maks poeng: 12

24 Shell scripting

We can automate tasks by using shell programming.

(1) Explain what the following script does. (6 points)

(2) What is the effect on line 09? (4 points)

```
01  #!/bin/bash

02  echo -e "Enter some names: \c"
03  read -a inputs

04  n=0
05  while [ $n -lt ${#inputs[@]} ]
06  do
07    if [ -d ${inputs[$n]} ]
08    then
09      chmod -x ${inputs[$n]}
10    fi
11    n=$(( n+1 ))
12  done
```

Fill in your answer here

You enter some names into this program. Then this program goes through all these names, and it gives all the names execution rights to their designated folder. <- This is what happens on the 9. line. It gives the specified user execution rights

Maks poeng: 10