Forside IDATG2202 Nov 2022

Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Eksamensoppgave i IDATG2202 Operativsystemer

Eksamensdato: 29.11.2022

Eksamenstid (fra-til): 09:00 - 12:00

Hjelpemiddelkode/Tillatte hjelpemidler: D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Faglig kontakt under eksamen: Erik Hjelmås
Tlf.: 93034446

ANNEN INFORMASJON:

Skaff deg overblikk over oppgavesettet før du begynner på besvarelsen din.

Les oppgavene nøye, gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensing av oppgaven. Faglig kontaktperson skal kun kontaktes dersom det er direkte feil eller mangler i oppgavesettet. Henvend deg til en eksamensvakt hvis du ønsker å kontakte faglærer. Noter gjerne spørsmålet ditt på forhånd.

- **Språk:** Alle oppgavetekster er på engelsk, men du står fritt til å svare på norsk eller engelsk eller "blanding".
- **Negative poeng/minuspoeng**: Ingen oppgaver kan føre til minuspoeng totalt på den respektive oppgaven, men noen av flervalgsoppgavene kan ha minuspoeng internt i oppgaven for å unngå at man "helgarderer". *Dette står da tydelig presistert i oppgaveteksten for de respektive oppgavene.*
- **Tolkning av spørsmål:** Hvis du er uenig i en oppgavetekst eller svaralternativene i en av de automatisk rettede oppgavene, så kommenter gjerne det i tekstfeltet i en av de vanlige tekst-svar-oppgavene.
- **Varslinger**: Hvis det oppstår behov for å gi beskjeder til kandidatene underveis i eksamen (f.eks. ved feil i oppgavesettet), vil dette bli gjort via varslinger i Inspera. Et varsel vil dukke opp som en dialogboks på skjermen i Inspera. Du kan finne igjen varselet ved å klikke på bjella øverst i høyre hjørne på skjermen.
- **Trekk fra/avbrutt eksamen:** Blir du syk under eksamen, eller av andre grunner ønsker å levere blankt/avbryte eksamen, gå til “hamburgermenyen” i øvre høyre hjørne og velg «Lever blankt». Dette kan ikke angres selv om prøven fremdeles er åpen.
- **Tilgang til besvarelse:** Etter eksamen finner du besvarelsen din i arkivet i Inspera. Merk at det kan ta én virkedag før eventuelle håndtegninger vil være tilgjengelige i arkivet.

Lykke til!

Hjelpeark OS

Se PDF.

1 introproc.os (2%)

What best describes the main function of an operating system?
Velg ett alternativ:

○ Allow user programs to directly control the CPU

○ [✓ Manage system resources and provide a set of services to user programs]

○ Allow user programs to manage system resources directly

○ Boot the system and hand over control of the keyboard and mouse to user programs

Maks poeng: 2

2 hwreview.register (2%)

Which register is used when addressing/referencing local variables in memory?
Velg ett alternativ:

○ EFLAG/RFLAG

○ EIP/RIP

○ [✓ EBP/RBP]

○ EAX/RAX

○ EBX/RBX

○ ESP/RSP

Maks poeng: 2

3 hwreview.assembly (2%)

```
mov $0, %eax
```

This assembly code does the following:

Velg ett alternativ:

○ Stores the contents of a register in memory address 0.

○✓ Stores 0 in a register.

○ Retrieves data from memory address 0 in RAM.

○ Renames variable $0 to %eax.

○ Moves the content of %eax to the variable $0.

Maks poeng: 2

4 hwreview.privileged (2%)

An example of a privileged instruction (a kernel mode instruction) is:

Velg ett alternativ:

○ Fetch the contents of a memory address into a register

○ Jump to a new memory location

○✓ Disable interrupts

○ Add the contents of two registers

Maks poeng: 2

5 introproc.os (2%)

How does a modern operating system make it possible to run multiple processes "simultaneously" (at the same time) on a CPU?

Velg ett alternativ:

○ By restricting the number of processes that are allowed to execute privileged instructions.

☒ Makes use of a hardware timer to provide limited time intervals to processes.

○ Waits for the running process to do I/O and then gives the CPU to the next process in the ready queue.

○ By interleaving (merging) instructions from all the processes in the ready queue.

Maks poeng: 2

6 introproc.mode (2%)

Which of the following C-statements would trigger a mode switch during program execution?

Velg ett alternativ:

○ char buffer[5];

○ eax=x*7;

○ if (x=0) {};

☒ fd=open("f.txt");

Maks poeng: 2

7 scheduling.rr (2%)

With the round robin scheduling algorithm:

Velg ett alternativ:

○ [✓ using very large time slices converts it into First In First Out scheduling algorithm]

○ using very small time slices converts it into Shortest Job First algorithm

○ using extremely small time slices increases performance

○ using very small time slices converts it into First In First Out scheduling algorithm

Maks poeng: 2

8 addrspace.pte (2%)

Which one of the following are a part of a Page Table Entry?

Velg ett alternativ

○ Local bit

○ Page bit

○ Virtual bit

○ [✓ Referenced bit]

Maks poeng: 2

9 addrspace.TLB (2%)

Which statement is correct about TLB?
Velg ett alternativ

○ TLB misses are the cause of Major Page Faults.

○ TLB points to the current page table.

○ TLB performs address translation.

[✓ ○ TLB contains copies of Page Table Entries.]

Maks poeng: 2

10 threads.vsproc (2%)

A main benefit of using threads for concurrency over processes is:
Velg ett alternativ:

○ Threads are faster when executing

○ Threads do not require any memory (stack, heap) during execution

○ Threads require less resources and overhead to create ✓

○ Threads do not need to be attached to a process

Maks poeng: 2

11 semaph.condvar (2%)

Which statement is correct about condition variables?
Velg ett alternativ:

○ A condition variable must always be used together with a semaphore.

[✓ ○ A condition variable does not contain a value.]

○ A binary condition variable can be used a mutex lock.

○ A condition variable that is zero will block processes/threads.

Maks poeng: 2

12 io.mmio (2%)

If we are using Memory-Mapped I/O, which assembly instruction can be used to reach the controller on an I/O-device?
Velg ett alternativ:

○ jmp

[✓ ○ mov]

○ in

○ out

Maks poeng: 2

13 fscore.chmod (2%)

Which part of the file system is written to when we do chmod 600 f.dat?

Velg ett alternativ:

○ Directory entry

○ Primary data block

✓ ○ Inode

○ Superblock

Maks poeng: 2

14 virt.container (2%)

Which operating system feature makes sure that when you are inside a container you cannot see the processes of the host operating system or other containers?

Velg ett alternativ

✓ ○ Namespacing.

○ Control groups.

○ Union mounts.

○ Address Space IDentifier (ASID).

Maks poeng: 2

15 ossec.NX (2%)

The NX-bit is located in

Velg ett alternativ:

○ First entry above the base pointer on stack.

○ The Instruction Register (IR).

○ The PSW/FLAG register.

○ [✓ Page Table Entries.]

Maks poeng: 2

16 introproc.states (2%)

What are the three most common states for a process? (a correct option gives X points, an incorrect option gives a penalty of minus 0.5X. In total you will not get fewer than zero points).

Velg ett eller flere alternativer:

□ Starting

☒ Ready

☒ Blocked on I/O

□ Hyperthreading

☒ Running

□ Busy waiting

□ Switching

□ Killed

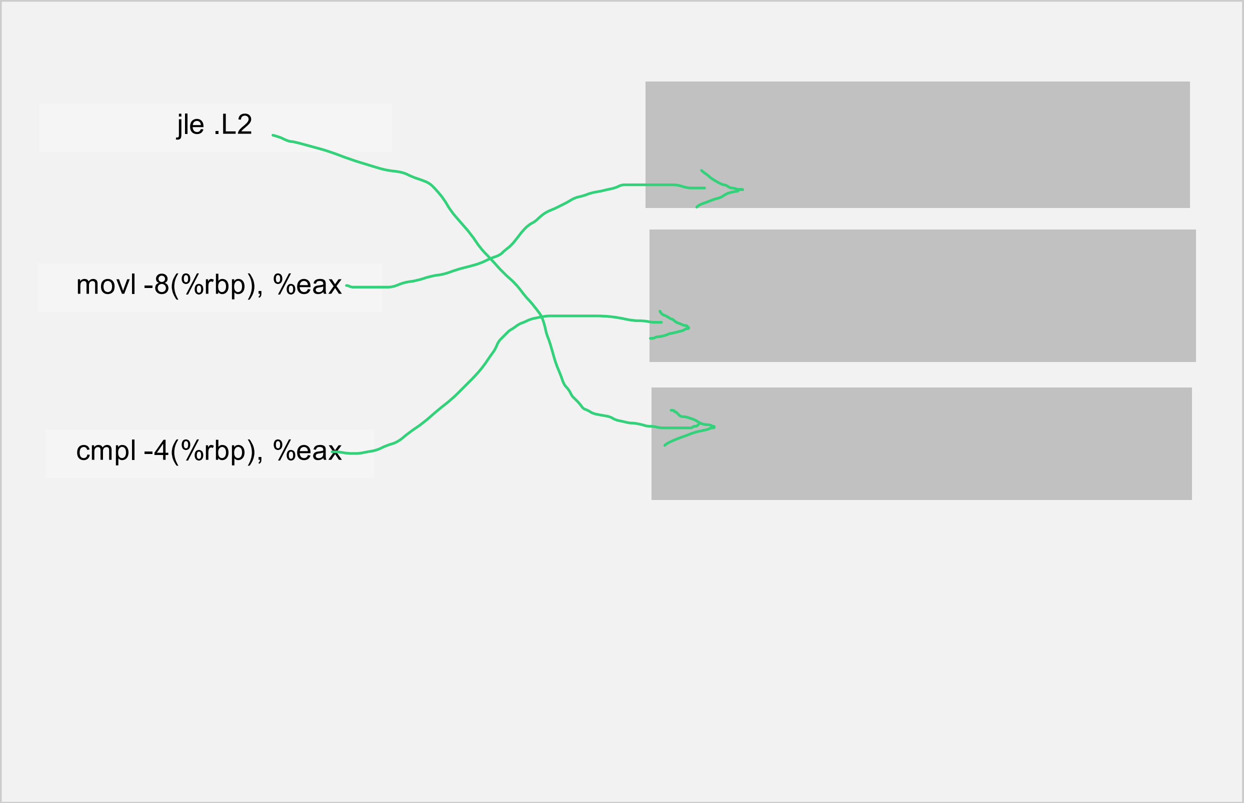
□ Thrashing

Maks poeng: 2

17 hwreview.assembly (3%)

The following three lines of assembly code (which logically belong together) are in the wrong order (when reading from the top like we do when we read code). Move the lines to the right in such a way that they end up in the correct order.

Dra og slipp



```
jle .L2

movl -8(%rbp), %eax

cmpl -4(%rbp), %eax
```

Maks poeng: 3

18 introproc.time (3%)

A single-threaded CPU-bound process uses six minutes when running on a single-core CPU. How much time will it take for five such processes to complete (when they start at the same time) on a modern preemptive multitasking operating system (such as Windows or Linux) on a quad-core CPU (quad-core is four cpu cores) ?

Velg ett alternativ:

○ 18 minutes

○ 12 minutes

○ 9 minutes

○ 7.5 minutes ??

○ 6 minutes

○ 4.5 minutes

○ 3 minutes

○ 2.5 minutes

○ 2 minutes

At the start, 4 processes run in parallel, taking 6 minutes.
Now the 5th can start, and since it has all the four cores to itself:
6min/4 cores = 1.5min
-> 6+1.5 min = 7.5min ✓

Maks poeng: 3

19 scheduling.MLFQ (6%)

MLFQ has the following rules

1. If Priority(A) > Priority(B), A runs (B doesn’t).
2. If Priority(A) = Priority(B), A and B run in Round Robin.
3. When a job enters the system, it is placed at the highest priority (the topmost queue).
4. If a job uses up an entire time slice while running, its priority is reduced (i.e., it moves down one queue).
5. If a job gives up the CPU before the time slice is up, it stays at the same priority level.
6. After time period S, move all the jobs in the system to the topmost queue.

Given the following setup on a system with a single CPU:

- Three queues Q0, Q1 and Q2 where Q2 is the highest priority queue
- Time slice for all queues are 10ms
- S is 50ms (priority boost every 50 ms)

The following processes arrive at time 0 in order P0, P1:

Process name	Run time	I/O frequency	I/O time
P0	25	5	5
P1	38	15	5

Note the following:

- If I/O frequency is N it means the process does I/O every N ms
- When I/O frequency and time is zero, it means the process does not do any I/O.
- Scheduling decision is made
    - When a job completes (exits)
    - When a job does I/O
    - When I/O for a job completes
    - When a job has used its time slice
    - When a priority boost happens
- A new time slice starts after every scheduling decision, unless a process is interrupted by a higher priority process, in that case the process stays at the head of the queue at its priority level and will resume to complete the rest of its time slice.

Use pen and paper to write down how these processes will run, then answer the following questions:

- When P0 completes, it exits queue Velg alternativ ⌄ ([✓ Q2], Q1, Q0)
- When P1 completes, it exits queue Velg alternativ ⌄ ([✓ Q2], Q1, Q0)
- In which queue is P1 at time 40? Velg alternativ ⌄ (Q2, [✓ Q1], Q0)
- Turn-around time for P1 (ms): Velg alternativ ⌄ (60, 63, 65, [✓ 68], 70, 72, 75, 78, 80, 83, 85, 90)
- Response time for P1 (ms): Velg alternativ ⌄ (0, [✓ 5], 10, 15, 20, [✓ 25], 30, 35, 40)
- During the first 50ms, how much CPU time is used by P1? Velg alternativ ⌄ (10ms, 15ms, 20ms, [✓ 25ms], 30ms, 35ms, 40ms)

Maks poeng: 6

20 addrspace.paging (6%)

We have a system with page-based address translation with 16-bit virtual addresses, 8-bit offset and the following part of the page table:

```
VPN  PFN       Present-bit
   +-----------+---+
.
.
.
15 | 0000 0110 | 1 |
14 | ~~0110 0000~~ | 1 |
13 | 0111 0111 | 1 |
12 | 1011 1011 | 1 |
11 | 0000 1110 | 1 |
10 | 0000 0000 | 0 |
9  | 0010 1010 | 1 |
8  | 0001 1111 | 1 |
7  | 0000 1011 | 1 |
6  | 0000 0001 | 1 |
5  | 1110 1000 | 1 |
4  | 0000 0000 | 0 |
.
.
.
   +-----------+---+
```

The page table has Velg alternativ (16, 32, 64, 128, [✓ 256], 512, 1024, 2048, 4096) entries.

The page size is Velg alternativ (64, [✓ 256], 1K, 4K) bytes.

The size of the virtual address space is Velg alternativ ([✓ 64K], 256K, 1K, 4K, 4M, 4G) bytes.

The virtual address 0000 1110 1010 1011 would be translated to Velg alternativ (0000 0110 1010 1011, [✓ 0110 0000 1010 1011], 0111 0111 1010 1011, 1011 1011 1010 1011, 0000 1110 1010 1011, 0000 0000 1010 1011, 1010 1011 1010 1011, 0001 1111 1010 1011, 1010 1011 1010 1011, 0000 0001 1010 1011, 1110 1000 1010 1011, 0000 0000 1010 1011)

Maks poeng: 6

21 syscalls.fork (12%)

Write a C-program mysil.c that will

1. (4%) Fork a child process and separate between the code that should be executed in the child process and the code that should be executed in the parent process.
2. (3%) The parent process should output (using printf) an address from the data/heap area in memory.
3. (3%) The child process should output (using printf) an address from the stack area in memory.
4. (2%) Write the command line you would use to compile mysil.c into the executable file mysil, include all warnings.

Skriv ditt svar her

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>
int main(int argc, char **argv) {
  pid_t rc = fork();
  if (rc < 0) {
    fprintf(stderr, "FORKING FAILED!\n");
  } else if (rc == 0) {
    printf("Addr of stack data: %p\n", main);
  } else {
    wait(NULL);
    printf("Addr of heap data: %p", malloc(100e6));
  }
}
```

Maks poeng: 12

22 semaph.sync (15%)

Consider the C-code:

```
int balance = 0;
void* worker(void* arg) {
    balance++;
    printf("%s made balance: %d\n", (char*) arg, balance);
    return NULL;
}
int main(void) {
    worker((void*) "Mainthread");
    return 0;
}
```

When executed, this prints:

```
$ ./a.out
Mainthread made balance: 1
```

1. (5%) Expand this code so it will also create a thread. The thread should also run the code in the worker function. The output could be (it might be in a different order or with other values since we have no synchronization yet):
2.
```
$ ./a.out
Mainthread made balance: 1
Newthread made balance: 2
```
3. (5%) Expand the code again with locks to make sure the final value of balance is guaranteed to be 2 and the output lines are correct.
4. (5%) Expand the code again to guarantee that Newthread will execute before Mainthread.

Skriv ditt svar her

```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>
#include <unistd.h>
sem_t sema;
int balance = 0;
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;
void *worker(void *arg) {
  pthread_mutex_lock(&lock);
  balance++;
  printf("%s made balance: %d\n", (char *)arg, balance);
  pthread_mutex_unlock(&lock);
  sem_post(&sema);
  return NULL;
}
int main(void) {
  pthread_t thread;
  sem_init(&sema, 0, 0);
  pthread_create(&thread, NULL, worker, (void *)"Newthread");
  sem_wait(&sema);
  worker((void *)"Mainthread");
  pthread_join(thread, NULL);
  return 0;
}
```

Maks poeng: 15

23 fscore.freespace (7%)

To keep track of free space in memory or in any data storage, the operating system makes use of either a freelist or a bitmap.

1. (2%) Briefly explain how a freelist works.
2. (2%) Briefly explain how a bitmap works.
3. (3%) Calculate the size of the bitmap in a page-based memory system with page size 2MB and physical memory of 64GB?

Skriv ditt svar her

Format

Free list works by having two values: The start of the address, and how many blocks/pages it occupies upwards.

Bitmap works by having one bit value indication if the page is free (1) or used (0)

64GB/2MB = 2^36 / 2^21 = 2^15b = 2^12 B = 4KB

Words: 0

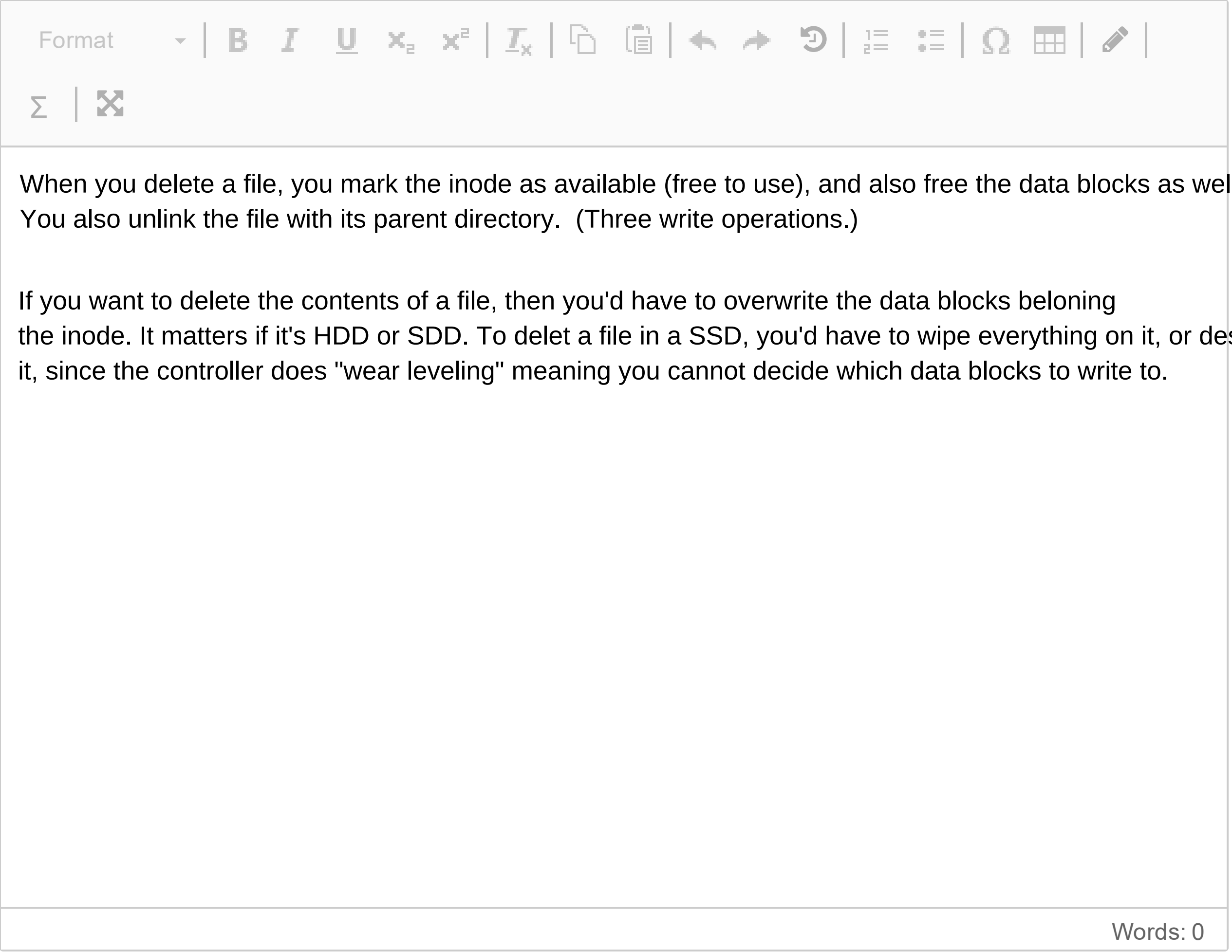
Maks poeng: 7

24 fscore.delete (7%)

You probably know that when you delete a file, you are not actually deleting the content of the file.

1. (3%) What happens in the file system when you delete a file.
2. (4%) What do you have to do if you want to delete the content of a file (a so-called "secure delete")? Does it matter if the underlying storage medium is a HDD or a SSD? Justify your answer.

Skriv ditt svar her



When you delete a file, you mark the inode as available (free to use), and also free the data blocks as well. You also unlink the file with its parent directory. (Three write operations.)

If you want to delete the contents of a file, then you'd have to overwrite the data blocks beloning the inode. It matters if it's HDD or SDD. To delet a file in a SSD, you'd have to wipe everything on it, or destroy it, since the controller does "wear leveling" meaning you cannot decide which data blocks to write to.

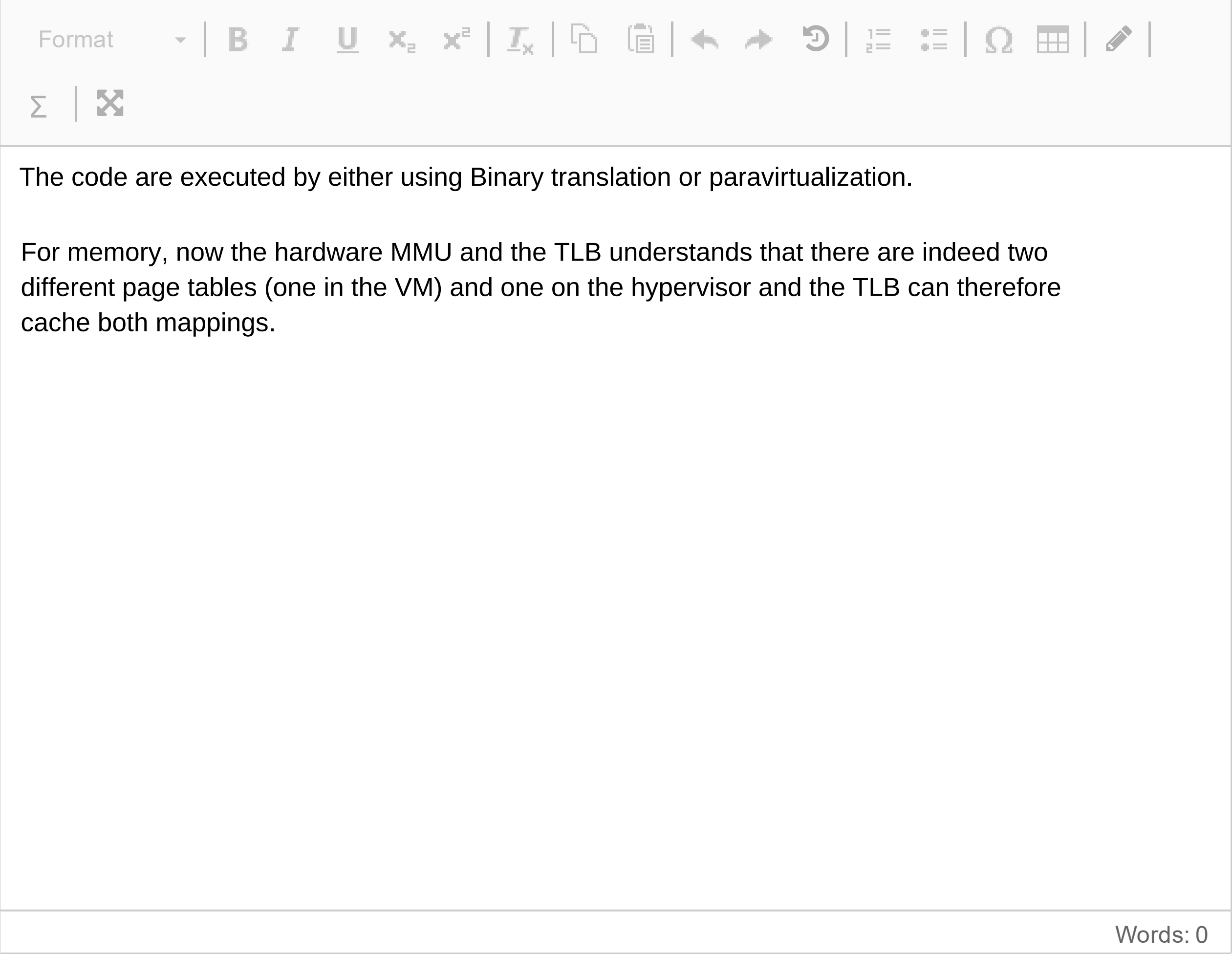
Maks poeng: 7

25 virt.vm (4%)

Running a virtual machine (VM) is just running an operating system and applications on "simulated hardware" right? No, not really.

Briefly explain what is different when it comes to *executing code on the CPU* and *using Memory* when the operating system and applications are running inside a virtual machine instead of natively on hardware.

Skriv ditt svar her



The code are executed by either using Binary translation or paravirtualization.

For memory, now the hardware MMU and the TLB understands that there are indeed two different page tables (one in the VM) and one on the hypervisor and the TLB can therefore cache both mappings.

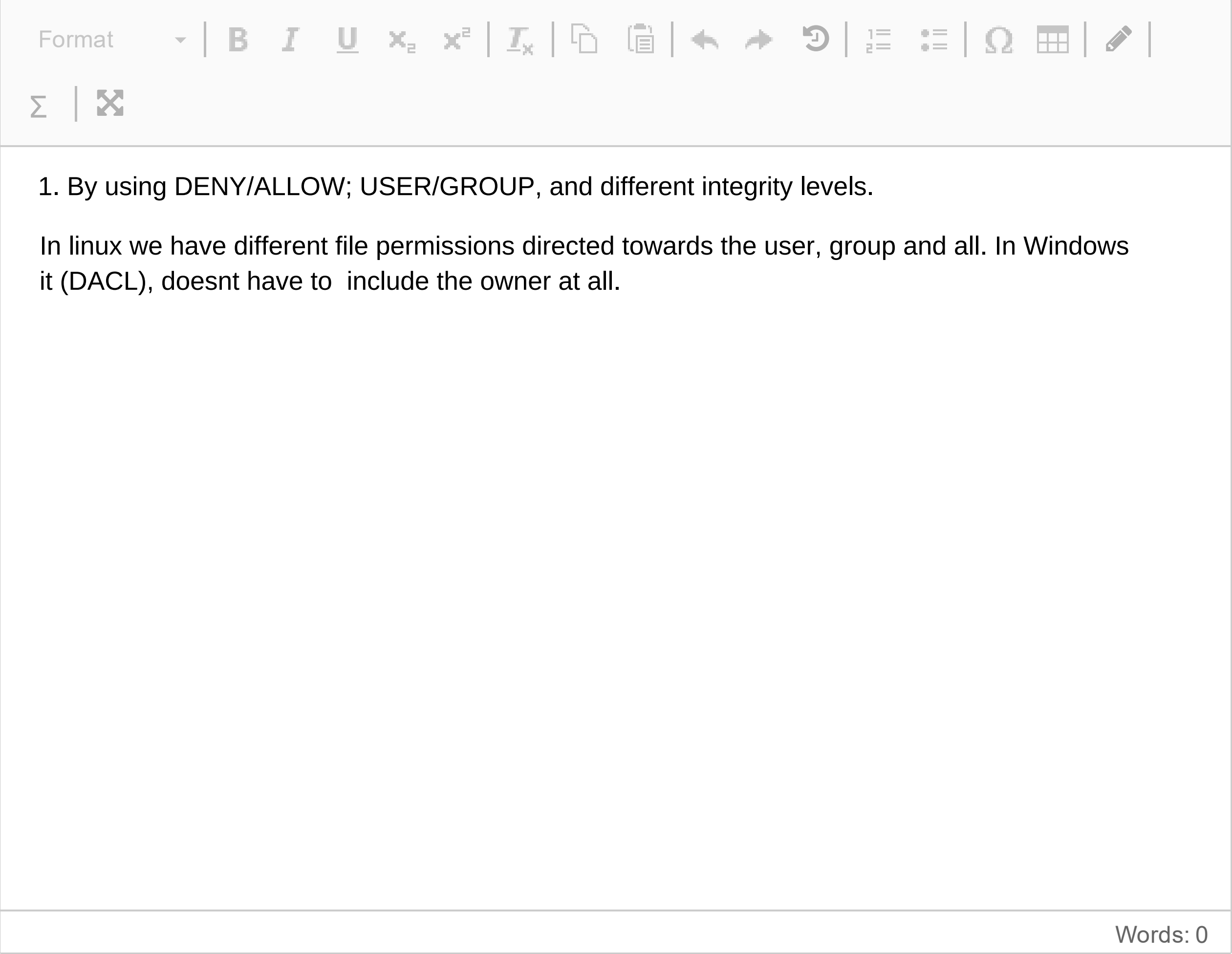
Maks poeng: 4

26 ossec.access (5%)

Access control in Windows and Linux:

1. (3%) Give an example of a discretionary access control list (DACL) in Windows. How can we override the DACL in Windows (in other words: which mechanism exists that take precedence over DACL).
2. (2%) Is the owner of a file used in access control in the same way in both Linux and Windows? Justify your answer.

Skriv ditt svar her



1. By using DENY/ALLOW; USER/GROUP, and different integrity levels.

In linux we have different file permissions directed towards the user, group and all. In Windows it (DACL), doesnt have to include the owner at all.

Maks poeng: 5

Document 2

Attached

Multiplication

$$2^n \cdot 2^m = 2^{n+m}$$

$$\text{e.g. } 2^{12} \cdot 2^{20} = 2^{12+20} = 2^{32} = 4G$$

Division

$$\frac{2^n}{2^m} = 2^{n-m}$$

$$\text{e.g. } \frac{2^{32}}{2^{12}} = 2^{32-12} = 2^{20} = 1M$$

Useful Tables

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1024$

$2^{10} = 1K$ (Kilo)
$2^{20} = 1M$ (Mega)
$2^{30} = 1G$ (Giga)
$2^{40} = 1T$ (Tera)
$2^{50} = 1P$ (Peta)
$2^{60} = 1E$ (Exa)
($2^{70} = 1Z$ (Zetta))
($2^{80} = 1Y$ (Yotta))

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Hex is sometimes prefixed 0x and binary is sometimes prefixed 0b.

We use b for bit and B for byte (a byte is eight bits). Each address into memory goes to a byte, NOT a bit. A byte is the smallest unit we can reference/address in memory.

ErikH, November 24, 2022