ℹ Forside IDATG2202 Nov 2023

Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Eksamensoppgave i IDATG2202 Operativsystemer

Eksamensdato: 28.11.2023

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

Faglig kontakt møter i eksamenslokalet: Nei.

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.*
- **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!

NYNORSK VERSJON:

Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Eksamensoppgåve i IDATG2202 Operativsystemer

Eksamensdato: 28.11.2023

Eksamenstid (frå-til): 09:00 - 12:00

Hjelpemiddelkode/Tillatne hjelpemiddel: **D**: Ingen trykte eller handskrevne hjelpemiddel tillate. Bestemt, enkel kalkulator tillate.

Fagleg kontakt under eksamen: Erik Hjelmås

Tlf.: 93034446

Fagleg kontakt kjem til eksamenslokalet: Nei.

ANNAN INFORMASJON:

Skaff deg eit overblikk over oppgåvesettet før du byrjar å svare på oppgåvene.

Les oppgåvene nøye, gjer deg opp dine eigne meiningar og presiser i svara dine kva for føresetnadar du har lagt til grunn i tolking/avgrensing av oppgåva. Fagleg kontaktperson skal berre kontaktast dersom du meiner det er direkte feil eller manglar i oppgåvesettet. Vend deg til ei eksamensvakt om du ynskjer å kontakte faglærar. Noter gjerne spørsmålet ditt på førehand.

- **Språk:** Alle oppgåvetekstar er på engelsk, men du står fritt til å svare på norsk eller engelsk eller "blanda".
- **Negative poeng/minuspoeng**: Ingen oppgåver kan føre til minuspoeng totalt på dei respektive oppgåva, men nokon av flervalgsoppgavene kan ha minuspoeng internt i oppgåva for å unngå at ein "heilgarderer". *Dette står då tydeleg presistert i oppgåveteksten for dei respektive oppgåvene*.
- **Varslingar**: Dersom det oppstår behov for å gje beskjedar til kandidatane medan eksamen er i gang (f.eks. ved feil i oppgåvesettet), vil dette bli gjort via varslingar i Inspera. Eit varsel vil dukke opp som en dialogboks på skjermen i Inspera. Du kan finne att varselet ved å klikke på bjølla i øvre høgre hjørne på skjermen.
- **Trekk frå/avbroten eksamen:** Blir du sjuk under eksamen, eller av andre grunnar ynskjer å levere blankt/avbryte eksamen, gå til “hamburgermenyen” i øvre høgre hjørne og vel «Lever blankt». Dette kan ikkje angrast sjølv om prøven framleis er open.
- **Tilgang til svara dine:** Etter eksamen finn du svara dine i arkivet i Inspera. Merk at det kan ta ein virkedag før eventuelle handteikningar vert tilgjengelege i arkivet.

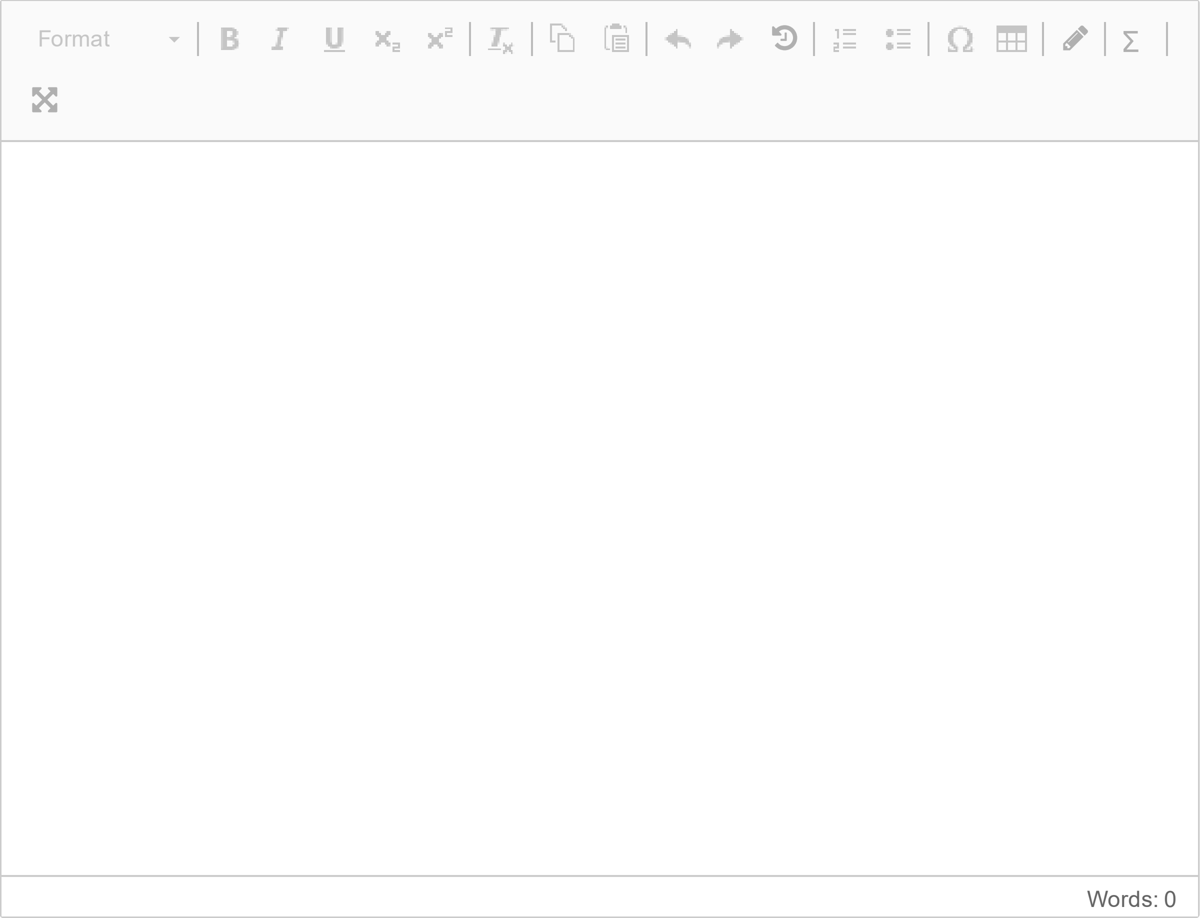
Lykke til!

1 Merknader

Bokmål:
Dette tekstfeltet kan du benytte om du i løpet av eksamen finner det nødvendig å legge inn egne kommentarer eller presiseringer vedrørende enkelte oppgaver. Husk å merke tydelig hvilken/hvilke oppgaver det er snakk om. (Feltet er ikke poenggivende i seg selv.)

Nynorsk:
Dette tekstfeltet kan du nytte om du i løpet av eksamen finn det nødvendig å leggje inn eigne kommentarar eller presiseringar vedkomande enkelte oppgåver. Hugs å merke tydeleg kva/korleis oppgåver det er snakk om. (Feltet er ikkje poenggjevande i seg sjølv.)

Dine merknader



Maks poeng: 0

2 introproc.os (2%)

What best describes the main function of an operating system?

Velg ett alternativ:

○ Allow user programs to manage system resources directly

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

○ Allow user programs to directly control the CPU

☒ Manage system resources and provide a set of services to user programs

Maks poeng: 2

3 introproc.mode (2%)

Which of the following C-statements would trigger a mode switch (a transition from user mode to kernel mode) during program execution?

Velg ett alternativ:

○ c++;

○ int x=5;

✓ printf("me\n");

○ char buffer[5];

Maks poeng: 2

4 hwreview.register (2%)

Which register contains the address to the next instruction that is going to be fetched from memory?

Velg ett alternativ:

○ EFLAG/RFLAG

○ EAX/RAX

○ EBP/RBP

○ EBX/RBX

○ ESP/RSP

[✓ ○ EIP/RIP]

Maks poeng: 2

5 hwreview.assembly (2%)

```
05          movq    %rsp, %rbp
06          movl    X, -4(%rbp)
07          cmpl    $4, -4(%rbp)
08          je     .L2
09          addl    $1, -4(%rbp)
10 .L2:
```

What should you replace **X** with in this assembly code to make this code skip (not execute) line number 09 ?

Velg ett alternativ:

○ 4(%rbp)

○ $-4

○ $0

○ -4(%rbp)

○ %eax

● $4

Maks poeng: 2

6 scheduling.theory (2%)

Which of the following criteria is more important for an *interactive* system?

Velg ett alternativ:

○ CPU utilization

○ Throughput

○ Turnaround time

● Response time

Maks poeng: 2

7 scheduling.theory (2%)

Which of the following scheduling algorithms gives the minimum average response time for a given set of processes?

Velg ett alternativ:

○ FIFO/FCFS

○ STCF/PSJF

○ SJF

○ [✓ RR]

Maks poeng: 2

8 addrspace.layout (2%)

Typical process address space (virtual address space) layout is (order: lowest to highest):

Velg ett alternativ:

○ [✓ Text/code, data, heap, stack (growing down towards heap)]

○ Text/code, data, stack (growing up towards heap), heap

○ Data, text/code, heap, stack (growing down towards heap)

○ Heap, stack (growing down towards heap), data, text/code

Maks poeng: 2

9 threads.theory (2%)

Which of the following items are *shared* across multiple threads belonging to the same process?
Velg ett alternativ:

○ [✓ Text/code, data/heap, files]

○ registers

○ stack

○ all of the above

Maks poeng: 2

10 semaph.sync (2%)

What does the pthread_cond_wait() call assume about the mutex that is passed as a parameter?
Velg ett alternativ:

○ It assumes that the mutex is initialized with the condition variable.

○ [✓ It assumes that the mutex is locked when pthread_cond_wait() is called.]

○ It assumes that the mutex is unlocked when pthread_cond_wait() is called.

○ It assumes that the mutex behaves like a semaphore.

Maks poeng: 2

11 io.ssd (2%)

What is a key challenge with Solid State Drives (SSD)?

Velg ett alternativ:

- ✓ you cannot overwrite data without erasing first
- ○ the flash translation layer (the controller) needs to update the inode whenever a file changes
- ○ using RAM as cache is not as efficient when the backend storage is SSD
- ○ larger blocks of random writes are slow

Maks poeng: 2

12 io.raid (2%)

How is the parity information distributed in a RAID 5 array?

Velg ett alternativ:

- ○ It is stored on a separate partition
- ✓ It is distributed across all the disks
- ○ It is not used in RAID 5
- ○ It is stored on a dedicated disk

Maks poeng: 2

13 fscore.access (2%)

With EXT/VSFS file system, and nothing present in any caches, how many disk accesses is required to access the content of the file /tmp/a.dat ?

Velg ett alternativ

R tmp a

○ 1

○ 3

○ 5

[✓ ○ 7]

○ 9

○ 11

○ 13

○ 15

Maks poeng: 2

14 fscore.intro (2%)

In the file system, a file is uniquely identified by its

Velg ett alternativ:

○ Hard link

○ File descriptor

[✓ ○ Inode number]

○ File name

Maks poeng: 2

15 virt.container (2%)

Which one of the following statements about containers is true?
Velg ett alternativ

○ Containers are isolated and protected based on the hypervisor interface.

○ Containers are isolated and protected based on the libc interface.

[✓ ○ Containers are isolated and protected based on the system call interface.]

○ Containers are isolated and protected based on the firmware interface.

Maks poeng: 2

16 ossec.acl (2%)

What happens if a user tries to access a file that has a (Discretionary) Access Control List (DACL) with both ALLOW and DENY entries for the user or the groups that the user belongs to?
Velg ett alternativ:

○ The user is denied access if the DENY entry is for the user and the ALLOW entry is for a group that the user belongs to.

○ The user is allowed access if the ALLOW entry is for the user and the DENY entry is for a group that the user belongs to.

[✓ ○ The user is always denied access.]

○ The user is always allowed access.

Maks poeng: 2

17 introproc.states (3%)

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:

- □ Switching
- □ Thrashing
- ☑ Ready
- □ Killed
- □ Busy waiting
- ☑ Blocked on I/O
- □ Starting
- □ Hyperthreading
- ☑ Running

Maks poeng: 3

18 introproc.time (4%)

A single-threaded CPU-bound process uses three minutes when running on a single-core CPU. How much time will it take for six 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:

4: 2:

3+1,5

○ 18 minutes

○ 12 minutes

○ 9 minutes

○ 7.5 minutes

○ 6 minutes

[✓ 4.5 minutes]

○ 3 minutes

○ 2.5 minutes

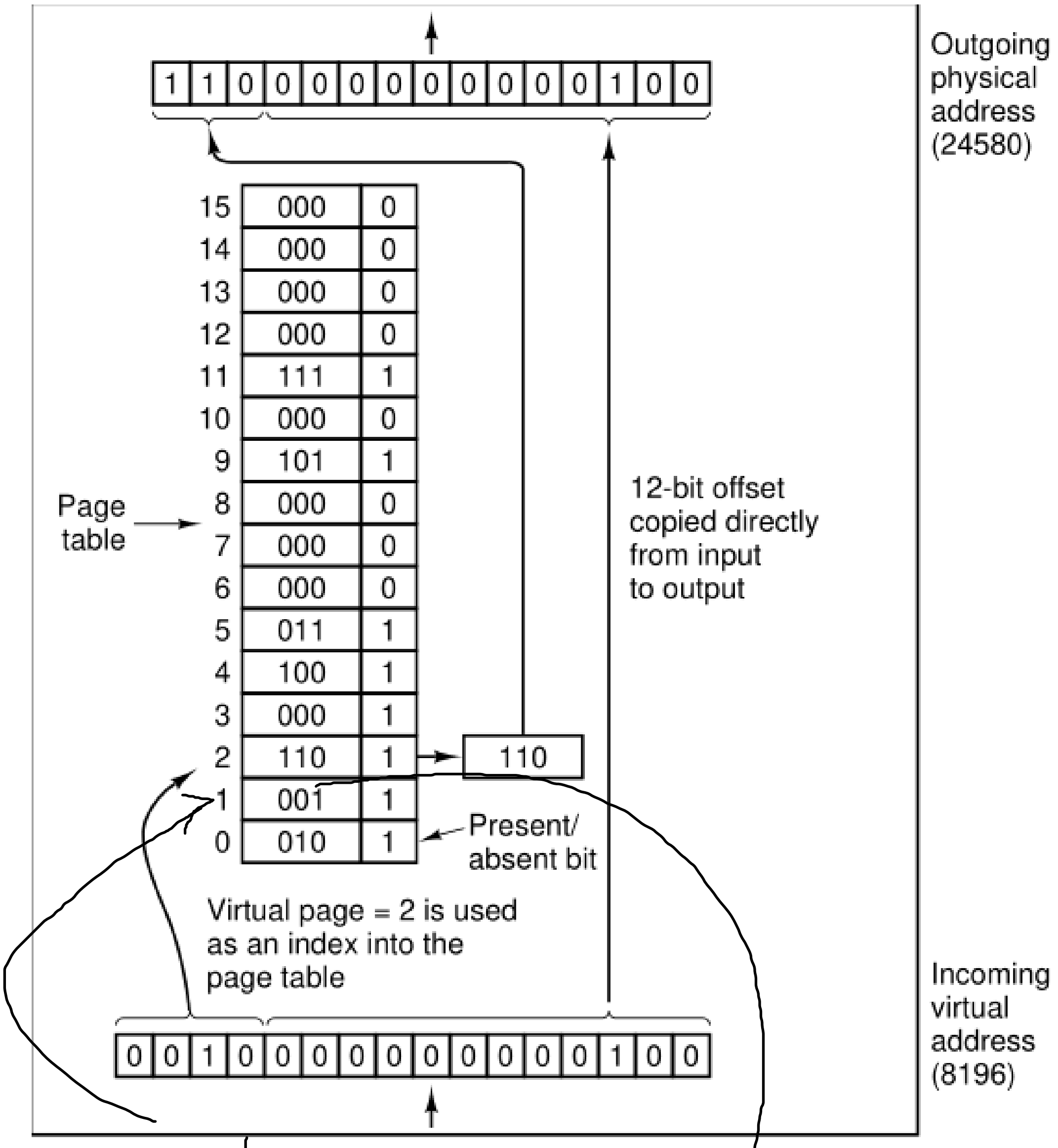
○ 2 minutes

Maks poeng: 4

19 addrspace.pagetable (4%)

Consider the following diagram where the incoming virtual address 8196 (decimal) is translated to the outgoing physical address 24580 (decimal).

If the incoming virtual address is 4098 (decimal), what will be the outgoing physical address (decimal)?



0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0

0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0

= 4098

Velg ett alternativ:

- ○ 24588
- ○ Gir page fault
- ○ 4108
- ○ 24580
- ○ 8204
- ○ 8196
- ○ 4096
- ○ 12
- ○ 24592
- ☒ 4098

Maks poeng: 4

20 fscore.dir (3%)

Which of the following items are present in a directory entry? (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:

□ Owner

□ Address to first data block

□ Permissions

☒ File name

□ Group(s)

☑ Inode number

□ Time stamps

□ File size

Maks poeng: 3

21 introproc.linux (4%)

At the bottom of the Inspera window there is a button that says “Linux-container”. When you press this for the first time, you will get an “APACHE GUACAMOLE” login window where you can either log in with your NTNU account or you can log in with

Username: gx

Password: Tada2day8

This will give you a terminal window to an enhanced Docker container which does not have internet access.

You can use this Linux-container during the exam to whatever you like, but be aware that if you save a file in Linux-container and then refresh or connect to Linux-container again, a new container will start and the file you have saved will be gone; the file system will look like the first time you connected.

It is important that you only have one Linux-container up at a time, otherwise the resources on the servers will be used up. Do not start multiple simultaneous containers.

You can copy text in Linux-container by marking it with the mouse and then pressing Shift-Ctrl-Alt. Then a clipboard with the marked text will appear and then you can copy with Ctrl-C and paste out here in the Inspera window with Ctrl-V. Internally inside Linux-container you can copy text by marking it with the mouse and then paste by right-clicking.

In the directory **mysil** you will find a program called **solan**. Execute this program and enter the output you get here: □

In the same directory you will also find a program called **melvind**. Add executable permissions to this program and execute it and enter the output you get here: □

Maks poeng: 4

22 fscore.linux (4%)

In the Linux-container, there is a directory **mx** with .txt-files and .pdf-files files, how many .txt-files are in this directory?

Velg ett alternativ:

○ 340

○ 344

○ 367

○ 374

○ 391

○ 395

○ 399

○ 415

○ 418

○ 431

○ 435

○ 436

○ 440

○ 446

Maks poeng: 4

23 syscalls.fork (9%)

Write a C-program fork.c that will

1. (3%) 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 child process must create a variable on the stack and output (using printf) the variables's value and the variable's address in memory.
3. (3%) The parent process must wait for the child process to exit before printing "Child done".

In the Linux-container you will find a file fork.c with the following contents:

```
#include <stdio.h>    /* printf */
#include <stdlib.h>   /* exit */
#include <sys/wait.h> /* wait */
#include <unistd.h>   /* fork */
int main(void) {
  pid_t rc=fork();
  // waitpid(PID,NULL,0);
  return 0;
}
```

You can use this file as starting point for your code. You do not have to include code for robustness (checking return codes for errors etc).

Note: You do not have to use the Linux-container to solve this problem but feel free to do so if you want to, but remember that the Linux-container is fragile: if you reload your container everything will disappear

Skriv ditt svar her

```
#include <stdio.h> /* printf */
#include <stdlib.h> /* exit */
#include <sys/wait.h> /* wait */
#include <unistd.h> /* fork */

int main(void) {
pid_t rc=fork();
if (rc == 0) {
  int x = 10;
  printf("Val: %d, addr: %p\n", x, &x);
  exit(0);
} else {
  wait(NULL);
   prinft("Child done"\n);
  return 0;
}
return 0;
}
```

Maks poeng: 9

24 semaph.sync (10%)

Consider the following C-program (you can also find this code in the file onetwothree.c in the Linux-container):

```
 1 void *ThreadOne(void *i) {
 2     printf("One!\n");
 3     return 0;
 4 }
 5
 6 void *ThreadTwo(void *i)  {
 7     printf("Two!\n");
 8     return 0;
 9 }
10
11 void *ThreadThree(void *i) {
12     printf("Three!\n");
13     return 0;
14 }
15
16 int main(void)
17 {
18     pthread_t tid1, tid2, tid3;
19     pthread_create(&tid1, NULL, ThreadOne, NULL);
20     pthread_create(&tid2, NULL, ThreadTwo, NULL);
21     pthread_create(&tid3, NULL, ThreadThree, NULL);
22     pthread_join(tid1, NULL);
23     pthread_join(tid2, NULL);
24     pthread_join(tid3, NULL);
25     return 0;
26 }
```

1. (4%) How many different combinations of one, two and three can this program output?
2. (6%) Explain (by entering which code you want to place in front of which line numbers) how you can use semaphores to guarantee that the printout will always be:
   One!
   Two!
   Three!

Skriv ditt svar her

1. 3! = 3*2*1 = 6 different combinations.

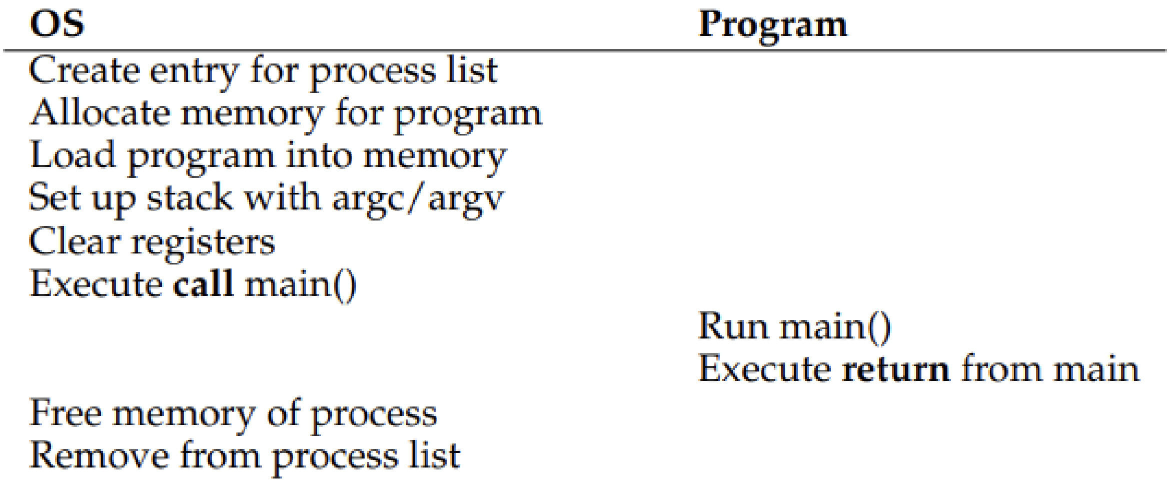
2. I would initalize the semaphore to starting value 0.
  Then i would sem_wait() before printing Two, sem_post() after printinf One.
Then sem_post after printing Two, sem_wait before printing Three.

Words: 0

Maks poeng: 10

25 syscalls.theory (5%)

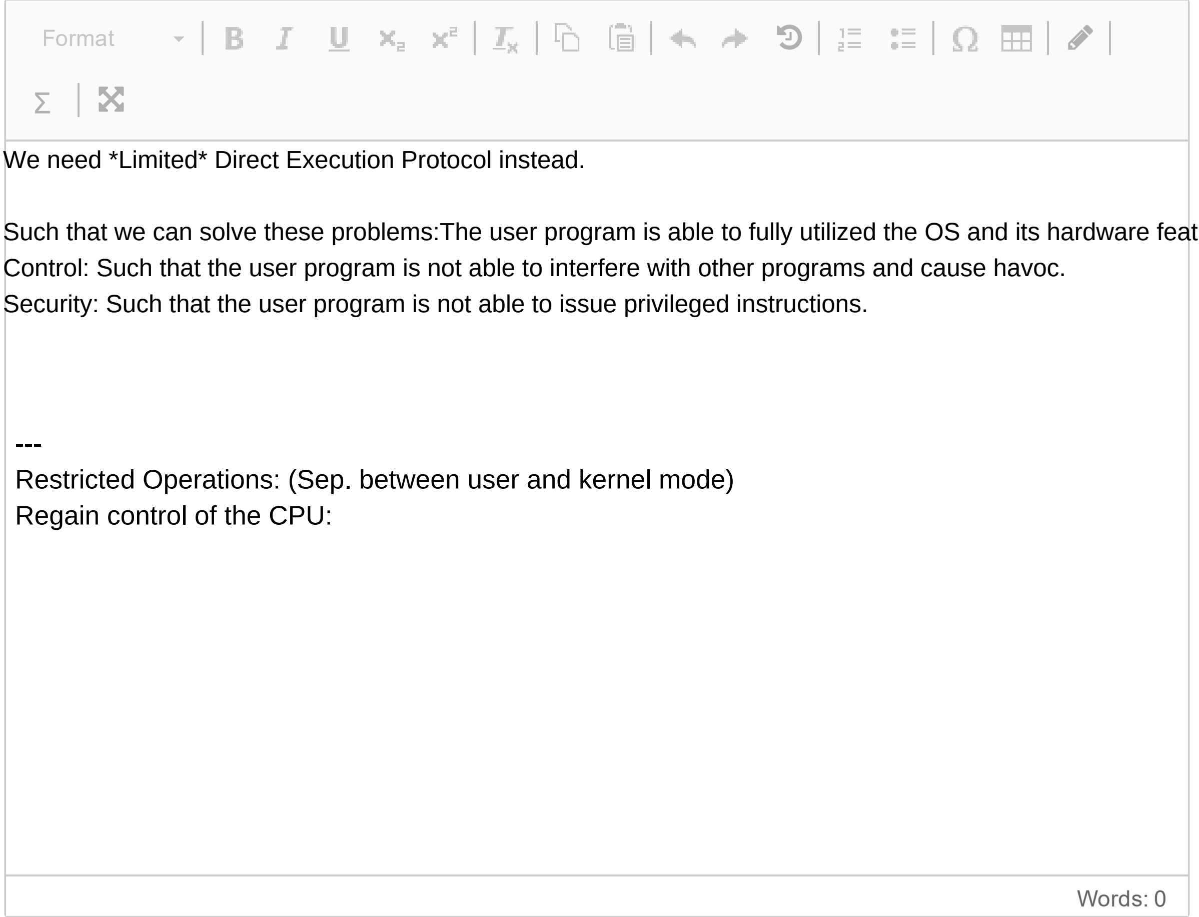
A computer system with hardware, operating system and user programs needs to be stable and reliable. Which two mechanisms are we missing in this figure for us to have a stable and reliable computer system?



OS	Program
Create entry for process list	
Allocate memory for program	
Load program into memory	
Set up stack with argc/argv	
Clear registers	
Execute **call** main()	
	Run main()
	Execute **return** from main
Free memory of process	
Remove from process list	

Figure 6.1: **Direct Execution Protocol (Without Limits)**

Skriv ditt svar her



We need *Limited* Direct Execution Protocol instead.

Such that we can solve these problems:The user program is able to fully utilized the OS and its hardware features.
Control: Such that the user program is not able to interfere with other programs and cause havoc.
Security: Such that the user program is not able to issue privileged instructions.

Restricted Operations: (Sep. between user and kernel mode)
Regain control of the CPU:

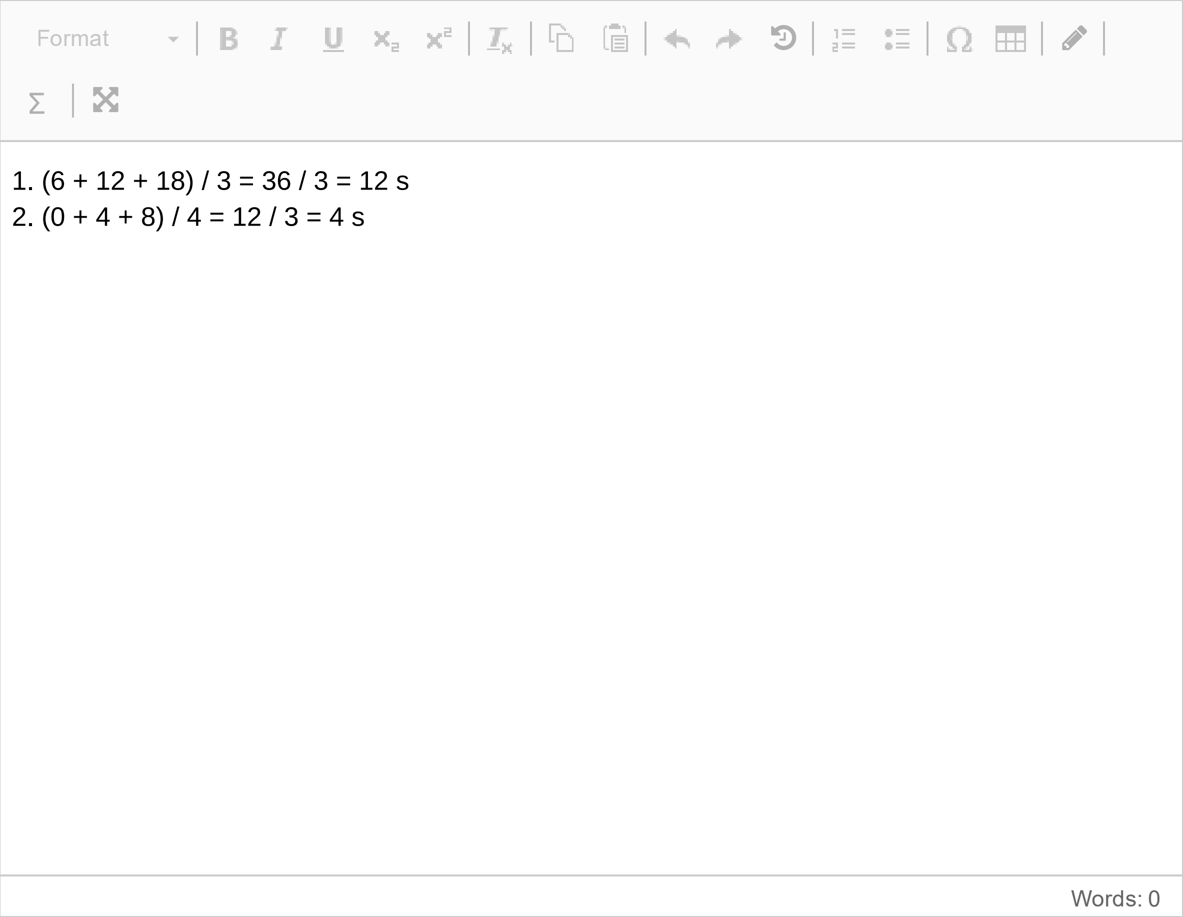
Maks poeng: 5

26 scheduling.time (6%)

Three processes arrive at time 0, and each require six seconds of CPU time (no I/O required).

1. (3%) Compute the average turnaround time when *FIFO (First-In First-Out)* is used.
2. (3%) Compute the average response time when *Round Robin with time slice four seconds* is used.

Skriv ditt svar her...



1. (6 + 12 + 18) / 3 = 36 / 3 = 12 s
2. (0 + 4 + 8) / 4 = 12 / 3 = 4 s

Maks poeng: 6

27 addrspace.bitmap (5%)

Calculate the size of the bitmap in a page-based memory system with page size 2MB and physical memory of 256GB? Give the answer in bytes, not bits.

Skriv ditt svar her

256GB / 2MB = 2^8*2^30/2^21 = 2^17b = 2^14 B = 16KB

Maks poeng: 5

28 memman.theory (6%)

1. (3%) When the memory management unit does address translation, which caching mechanism is involved specifically for address translation?
2. (3%) If the currently running process accesses/references bytes in memory one-by-one (a huge char-array) in a "cache-optimal way", how often will the *address translation caching mechanism* result in a cache miss (assuming you start with an empty cache)?

Skriv ditt svar her

Format

1. In that case, it's the Translation Look-Aside Buffer responsible for this.
2.

Words: 0

Maks poeng: 6

29 ossec.setuid (7%)

Consider the following setuid root executable and corresponding source code "adminshell.c". The program verifies a password before dropping the user to a root shell so that they can administrate the system.

```
$ ls -l adminshell
-rwsr-xr-x 1 root root 16208 Nov 8 12:53 adminshell

$ cat adminshell.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define PASSWORD "j935uhxxxjiAA"

int main(void) {
    int is_admin = 0;
    char password[32];

    // Get password from standard input
    gets(password);

    // Verify the password
    if(strcmp(password, PASSWORD) == 0) {
        printf("Password OK\n");
        is_admin = 1;
    } else {
        printf("Password failed\n");
    }

    if(is_admin) {
        // Spawn a root shell for the administrator
        setuid(0);
        setgid(0);
        system("/bin/sh");
        exit(0);
    }
}
```

Ignoring the fact that the plain-text password is present in the source code (and thus also in the binary executable), answer the following questions:

What is the security weakness? How can a malicious user that does not know the password abuse the program? Provide as many details as possible.

Skriv ditt svar her

First issue that i observe is the fact the use of the `gets` function which is NOT recommended to use at all, due to security reasons; that is, bufferoverflow attacks are rather easy to conduct.

The hacker can thus overflow the input to overwrite the `is_admin` variable and gain root access to the shell via `/bin/sh`.

Maks poeng: 7