

i Eksamensinformasjon

Fakultet: Teknologi, kunst og design

Utdanning: Teknologiske Fag

Emnenavn: Operativsystemer

Emnekode: DATA2500

Bokmål

Dato: 13 august 2021

Tid: kl. 09.00-12.00 (3 timer)

Hjelpemidler:

Alle hjelpemidler er tillatt.

Andre opplysninger:

Les nøye gjennom oppgavene før du begynner og pass på å besvare alle spørsmålene.

Sett gjerne egne forutsetninger dersom du synes oppgaveteksten er uklar. Beskriv forutsetningene og løs oppgaven utifra dem.

Maks poeng på de fleste av deloppgavene er 10 og de bidrar 3.33 % hver til sluttpoengsummen som maksimalt er 300. Men legg merke til at det er noen unntak:

Oppgave 5 og 6 gir inntil 5 poeng

Oppgave 13, 18, 19 og 20 gir inntil 15 poeng

Oppgave 23 gir inntil 50 poeng

Support:

Har du spørsmål til oppgaven under eksamen må du ta kontakt med Hårek Haugerud på denne Zoom-linken:

<https://oslomet.zoom.us/j/62852466651?pwd=QXYxSFNpbk85OEtyQ2xJTmxPbmNoZz09>

Meeting ID: 628 5246 6651

Password: 145965

Har du spørsmål om eller problemer med Inspira ta kontakt med eksamen-tkd@oslomet.no eller på telefon 67 23 88 88. Vi oppfordrer at kun akutte henvendelser kommer inn på telefon da denne har begrenset bemanning. Epost henvendelser vil bli behandlet fortløpende, husk å skrive inn studentnummer, kandidatnummer og emnekode i mailen for raskere behandling.

Opplever du problemer med innleveringen må du ta kontakt med eksamenskontoret **før innleveringsfristen går ut.**

1 Linux kommandolinje

En Linux-kommando kan motta en strøm av input fra en annen kommando gjennom

Velg ett alternativ:

- ☐ en daemon
- ☐ en environment variabel
- ☐ et signal
- ☐ et flagg
- ☐ en pipe
- ☐ en parameter
- ☐ en cmdlet
- ☐ en apostrof

Maks poeng: 10

2 Linux-kjernen

Hvilken av de følgende oppgavene utføres IKKE av Linux-kjernen?

Velg ett alternativ:

- ☐ Administrere ressursene slik at prosesser og brukere ikke ødelegger for hverandre.
- ☐ Gi applikasjonsprogrammer og brukere enhetlig, enklere og mer abstrakt adgang til maskinens ressurser.
- ☐ Lage et GUI som gjør at en bruker kan starte applikasjoner ved å klikke på ikoner.
- ☐ Håndtere internminnet (RAM) slik at prosesser ikke skriver over hverandres deler av minnet.
- ☐ Fordele CPU-tid mellom prosesser.
- ☐ Lage et filsystem som gir brukerne adgang til logiske filer.

Maks poeng: 10

3 Én prosess på to CPUer

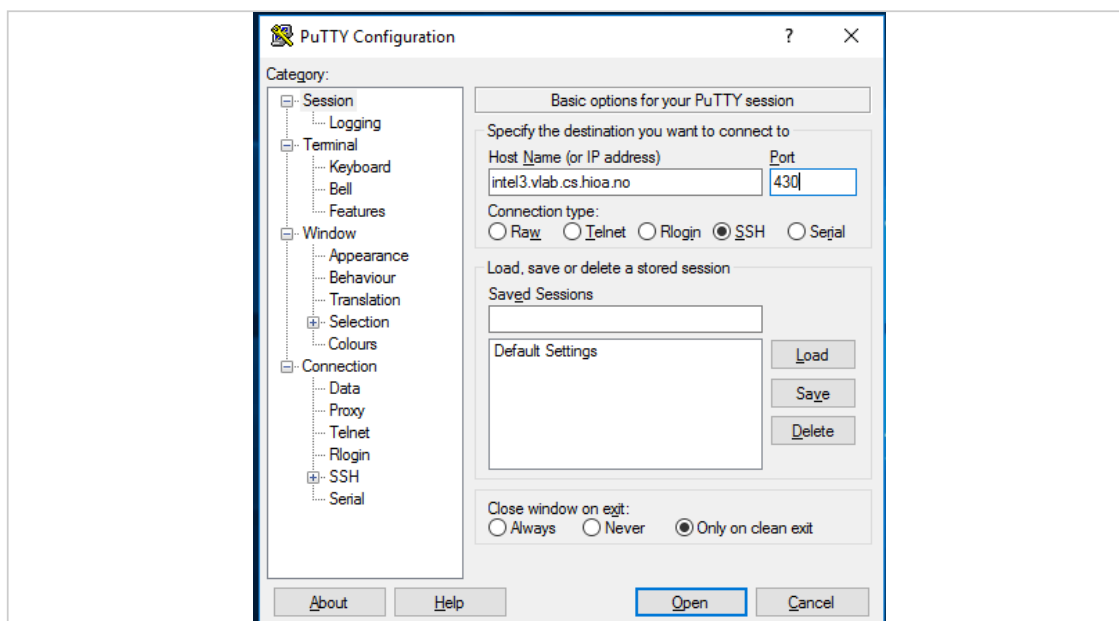
Hvorfor kan ikke OS få én vanlig sekvensiell prosess til å gå fortere ved å kjøre den på to CPUer?

Velg ett alternativ:

- ☐ Fordi en deadlock mellom CPUene da lett kan oppstå.
- ☐ Fordi OS ikke kjenner den indre logikken i prosessen og må la instruksjonene kjøres sekvensielt.
- ☐ Fordi OS ikke kan hente ut instruksjoner fra samme sted i RAM og kjøre det på to CPUer.
- ☐ Fordi OS ikke kan flytte en prosess fra en CPU til en annen etter at prosessen har startet.
- ☐ Fordi et OS kun styrer en CPU av gangen. Flere CPUer krever flere OS.
- ☐ Fordi to prosesser ikke kan ha samme navn.

Maks poeng: 10

4 Innlogging på Linux-VM



På StudentWeb har du fått oppgitt et kandidatnummer, det kan for eksempel være et tall som 430 og vi vil bruke dette tallet som eksempel, men bytt ut 430 med ditt eget kandidatnummer når du gjør disse oppgavene. Det samme kandidatnummeret står øverst i venstre i vinduet når du starter eksamen i Inspira.

På serveren intel3.vlab.cs.hioa.no har hver kandidat sin egen Linux-VM som man som en del av eksamen skal logge inn på for å løse oppgaver (disse er helt like Linux-VMene som dere brukte under kurset, med navn som os13, og de er også egentlig Docker-containere). Alle disse Linux-VMene har forskjellig innhold, slik at svaret på alle oppgavene vil være forskjellig for hver kandidat. For eksempel vil svaret på et spørsmål som "Totalt hvor mange filer finnes det på din Linux-VM" variere for hver kandidat, fordi det ikke ligger samme antall filer på hver VM.

Anta at du er kandidat nummer 430. Da må du først laste ned riktig passord fra denne web-siden:

<https://www.cs.hioa.no/~haugerud/os.php>

Skriv inn kandidatnummeret (ditt eget og ikke 430) og trykk på 'Send inn nummeret'-knappen. Hvis du skriver inn et gyldig kandidatnummer får du da et passord som er av samme type som for os-VMene og kan være noe lignende som **bien6tuft**. Det er viktig at du bare henter passord for din egen VM, hvilken IP som henter hvilket passord blir loggført. Det er viktig at du ikke går inn på andre kandidaters servere.

Logg så inn på din VM med kommandoen

```
ssh -p 430 kan430@intel3.vlab.cs.hioa.no
```

hvis du logger inn fra et bash-shell (Mac eller Linux) på din hjemme-PC og bruk så passordet du fikk fra websiden os.php på den samme hjemme-PCen (se instruksjoner for putty fra Windows lenger ned). Opsjonen -p 430 betyr at du bruker port-nummer 430 og ikke standard ssh-portnummer som er 22. Dette er laget slik for at hver kandidat skal nå sin egen VM. kan430 er brukernavnet du logger inn med. For både port-nummeret og brukernavnet må du bytte ut 430 med ditt eget kandidatnummer. Denne brukeren (kan430 i dette eksempelet) vil i senere oppgaver bli omtalt som "kan-brukeren".

Hvis du logger deg på direkte fra putty (se figur) må du bruke Host Name intel3.vlab.cs.hioa.no, skrive inn ditt kandidatnummer som port number (430 i eksempelet), bruke brukernavn kan430 og passordet du lastet ned (igjen, bytt ut 430 med ditt kandidatnummer).

For å svare på denne oppgaven, skriv inn følgende:

Kandidatnummer

Passordet du hentet fra websiden os.php

"Din IP-adresse" som du fikk fra websiden os.php

Innholdet av filen **file1** som ligger øverst i hjemme-mappen til din bruker (kan430)

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x























Words: 0

Maks poeng: 10

5 Fil på Linux-VM

I filesystemet til brukeren du logget på med på Linux-VM ligger det en tekst-fil med navn **file2**. Finn denne filen, kopier og paste ut innholdet av filen her:

Skriv ditt svar her

Format ▾ | **B** *I* U x_2 x^2 | I_x |   |    |   |   |  | Σ |



Words: 0

Maks poeng: 5

6 xfile på Linux-VM

I hjemme-mappen til brukeren root ligger det en fil med navn **xfile**. Kopier og paste ut innholdet av denne tekstfilen her:

Skriv ditt svar her

Format ▾ | **B** *I* U x_2 x^2 | I_x |   |    |   |   |  | Σ |



Words: 0

Maks poeng: 5

7 Grupper

Hvilke Linux-grupper er kan-brukeren (for eksempel "kan430") på din VM medlem av?
Vis også hvilken kommando du bruker for å finne ut dette.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x























Words: 0

Maks poeng: 10

8 Få root-tilgang

Når du er logget inn som en vanlig bruker (kan-brukeren, for eksempel "kan430") på VMen kan du bli root-bruker uten å kunne root-passordet (det er ikke engang satt et passord for root). Prøv å finne ut hvilken linje i konfigurasjonsfilen **/etc/sudoers** som gjør dette mulig og hvordan hvilke grupper som denne brukeren tilhører spiller en rolle for at kan-brukeren skal kunne bli root.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x

Words: 0

Maks poeng: 10

9 root uten passord

Når du er logget inn som en vanlig bruker (kan-brukeren, for eksempel "kan430") på VMen kan du på denne VMen bli root-bruker uten å engang skrive ditt eget passord (slik du måtte for VMene som ble brukt som OS-lab). Finn linjen i konfigurasjonsfilen **/etc/sudoers** som gjør at du ikke må taste eget passord for at kan-brukeren skal kunne bli root og skriv den inn her.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x

























Words: 0

Maks poeng: 10

10 Passord-hash og salt

Finn ut hvilken streng av tegn som er saltet (salt) som er brukt for å lage passord-hash'en til kanbrukeren på VMen.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x





















Σ



Words: 0

Maks poeng: 10

11 Lage hash-streng

Skriv ned en Linux-kommando som lager hele hash-strengen (inkludert salt) for passordet til kan-brukeren. (*Test at kommandoen virker på VMen*)

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x

Σ

Words: 0

Maks poeng: 10

12 Docker container

Inne på Linux-VM kjører en Docker-container. Finn ut hva dens container-ID og navn er og skriv inn begge her:

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x

























Σ



Words: 0

Maks poeng: 10

13 Koble til Docker-container

Inne i den kjørende Docker-containeren er det en fil i mappen **/root** som heter **xfile2**. Koble deg til containeren, finn filen inne på containeren og paste ut innholdet her:

Skriv ditt svar her

Format ▾ | **B** *I* U x_e x^2 | I_x |   |    |   |   |  |

Σ | 

Words: 0

Maks poeng: 15

14 Assembly

Assemblykoden

mov \$0, %eax

gjør følgende:

Velg ett alternativ:

- ☐ Lagrer variabelen \$0 i et register.
- ☐ Lagrer 0 i internminnet.
- ☐ Flytter innholdet av %eax til variabelen \$0.
- ☐ Lagrer 0 i et register.
- ☐ Lagrer innholdet fra et register i minneadresse 0.
- ☐ Endrer navn på variabelen \$0 til %eax.
- ☐ Henter data fra adresse 0 i internminnet.

Maks poeng: 10

15 Assembly

Assemblykoden

mov -4(%rbp), %eax

gjør følgende:

Velg ett alternativ:

- ☐ Henter data fra et register og legger i et annet register.
- ☐ Ganger verdien i registeret %rbp med -4 og legger resultatet i registeret %eax.
- ☐ Henter data fra cache og legger i et register.
- ☐ Henter data fra internminnet og legger i cache.
- ☐ Trekker -4 fra verdien i registeret %rbp og legger resultatet i registeret %eax.
- ☐ Henter data fra internminnet og legger i et register.

Maks poeng: 10

16 Prosesser

Hvilken Linux-kommando lister prosesser?

Velg ett alternativ:

- ☐ grep
- ☐ Ctrl-Alt-Del
- ☐ ping
- ☐ Get-Proc
- ☐ ls
- ☐ proc
- ☐ list
- ☐ ps

Maks poeng: 10

17 CPU-migration og context-switches

På en Linux-server med to CPUer (cores, regneenheter) har du et program **regn** som er 100% CPU-intensivt og hele tiden bruker så mye CPU-tid det kan. Du bruker så programmet perf og kjører **regn** på følgende måte:

```
Linux$ perf stat -B -e cpu-migrations,context-switches ./regn
Performance counter stats for './regn':
      0          cpu-migrations
     52          context-switches
15,205851971 seconds time elapsed
```

Forklar kort hva output betyr, spesielt hva cpu-migrations og context-switches er og hva verdiene 0 og 52 betyr i dette tilfellet.

Skriv ditt svar her

Format

B

I

U

x_e

x^2

I_x

Σ

Words: 0

Maks poeng: 10

18 CPU-migration og context-switches

På samme Linux-server med to CPUer (cores, regneenheter) lager du et script **perf3.sh** som bruker **perf** til å kjøre flere instanser av **regn** på følgende måte:

```
Linux$ cat perf3.sh
#!/bin/bash
perf stat -B -e cpu-migrations,context-switches ./regn&
perf stat -B -e cpu-migrations,context-switches ./regn&
perf stat -B -e cpu-migrations,context-switches ./regn&
```

Når du så kjører dette scriptet får du følgende output:

```
Linux$ ./perf3.sh
Performance counter stats for './regn':
      23          cpu-migrations
    1 255          context-switches
22,695675133 seconds time elapsed

Performance counter stats for './regn':
      20          cpu-migrations
    1 240          context-switches
22,730097770 seconds time elapsed

Performance counter stats for './regn':
      22          cpu-migrations
    1 222          context-switches
22,916894095 seconds time elapsed
```

Forklar kort hva output betyr og hvordan resultatene kan tolkes. Forklar spesielt

1. Hvordan resultatene sier noe om hvordan Linux schedulerer tre CPU-intensive prosesser som kjører på 2 CPUer.
2. Hvorfor det er så mange context-switches sammenlignet med når kun en regn-jobb ble kjørt alene i forrige oppgave.
3. Hvordan kan tiden de tre prosessene bruker på å fullføres (litt mer enn 22,5 sekunder) beregnes med utgangspunkt i at en regn-jobb brukte 15,2 sekunder når den kjørte alene i forrige oppgave på samme server.

Skriv ditt svar her

Format
|
B
I
U
x₂
x²
I_x
|
📄
📄
|
↶
↷
↺
|
≡
≡
|
Ω
📱
|
✎

Σ
|
✖

Words: 0

Maks poeng: 15

19 CPU-migration og context-switches

På samme Linux-server med to CPUer (cores, regneenheter) og med det samme scriptet **perf3.sh**

```
Linux$ cat perf3.sh
#!/bin/bash
perf stat -B -e cpu-migrations,context-switches ./regn&
perf stat -B -e cpu-migrations,context-switches ./regn&
perf stat -B -e cpu-migrations,context-switches ./regn&
```

kjører du nå scriptet på følgende måte:

```
Linux$ taskset -c 0 ./perf3.sh
Performance counter stats for './regn':
              0      cpu-migrations
             1 984      context-switches
45,229327576 seconds time elapsed

Performance counter stats for './regn':
              0      cpu-migrations
             1 973      context-switches
45,234486634 seconds time elapsed

Performance counter stats for './regn':
              0      cpu-migrations
             1 971      context-switches
45,369822226 seconds time elapsed
```

Forklar kort hva output betyr og hvordan resultatene kan tolkes. Forklar spesielt

1. Hvorfor det nå ikke blir noen cpu-migrations.
2. Hvorfor det nå blir enda flere context-switches sammenlignet med når **perf3.sh** ble kjørt uten **taskset** i forrige oppgave.
3. Hvordan tiden de tre prosessene bruker på å fullføres (litt mer enn 45 sekunder) kan beregnes med utgangspunkt i at en regn-jobb brukte 15,2 sekunder når den kjørte alene i første oppgave på samme server.

Skriv ditt svar her

Format
|
B
I
U
x₂
x²
I_x
|
📄
📋
|
↶
↷
↺
|
☰
☷
|
Ω
🔢
|
✎

Σ
|
✖

Words: 0

Maks poeng: 15

21 Internminne

Hva er betydningen av begrepet **Random Access Memory**?

Velg ett alternativ

- ☐ At det er tilfeldig hvilken byte som hentes først om to byte leses samtidig
- ☐ At data som lagres kan bli lagt hvor som helst i minnet
- ☐ At det går like fort å skrive til minnet som å lese fra minnet
- ☐ At adressene til hver byte i minnet allokeres tilfeldig av operativsystemet
- ☐ At alle enheter har tilgang til minnet
- ☐ At det tar like lang tid å hente en byte fra hvor som helst i minnet

Maks poeng: 10

22 Program-segmenter

Hva er de tre segmentene et program i internminnet vanligvis består av?

Velg ett alternativ:

- ☐ Heap, Map og variabler
- ☐ Kode, instruksjoner og cache
- ☐ Data, variabler og MMU
- ☐ Map, cache og Text
- ☐ Kode, Data og Stack
- ☐ Stack, Heap og DLL.

Maks poeng: 10

23 Lage mapper og filer

I denne oppgaven skal du laget et bash-script **mapper.sh** som lager en mappestruktur og legger filer i disse mappene. For å kunne legge tilfeldige ord inn i filene som lages har du en fil som heter **ord.txt** og som inneholder en rekke vanlige ord som er fire bokstaver lange. Ved hjelp av kommandoen **shuf** kan du trekke ut tilfeldige ord på følgende måte:

```
Linux$ shuf -n1 ord.txt
vits
Linux$ shuf -n1 ord.txt
pund
Linux$ shuf -n1 ord.txt
siam
```

Utgangspunktet for mappesturkturen du skal lage er en fil **pw** med 150 passord som har blitt generert. Hvis du ser på de tre første linjene av denne filen, ser den slik ut:

```
Linux$ head -n 3 pw
agat9your
aide9yens
ajax3yank
```

For hvert av disse passordene skal du lage en mappe og legge inn tre filer med navn **file1**, **xfile** og **rett.txt**. Hvis du krymper pw-filen og bare har de tre første passordene i den, skal scriptet etter kjøringen ha laget mapper og filer med innhold som vist av følgende kjøring av scriptet:

```
Linux$ ./mapper.sh
Linux$ tree ex
ex
├── ex500
│   ├── file1
│   ├── rett.txt
│   └── xfile
├── ex501
│   ├── file1
│   ├── rett.txt
│   └── xfile
└── ex502
    ├── file1
    ├── rett.txt
    └── xfile
3 directories, 9 files
Linux$ cat ex/ex*/file1
500 agat9your
501 aide9yens
502 ajax3yank
Linux$ cat ex/ex*/xfile
dask 500
gubb 501
dept 502
Linux$ ls -l ex/ex*/rett.txt
--w--wx-wx 1 haugerud haugerud 0 aug. 4 20:51 ex/ex500/rett.txt
-rwxr-x--- 1 haugerud haugerud 0 aug. 4 20:51 ex/ex501/rett.txt
---xr----- 1 haugerud haugerud 0 aug. 4 20:51 ex/ex502/rett.txt
```

Som man ser av output skal det lages en hovedmappe med navn **ex** og for hvert passord i filen **pw** skal det lages en mappe med navn **ex500** for det første passordet, **ex501** for det neste, og så videre. I hver mappe skal det legges en fil med navn **file1** og denne skal inneholde mappennummeret etterfulgt av passordet. I tillegg skal det lages en fil med navn **xfile** som skal inneholdet et tilfeldig ord hentet fra filen **ord.txt** etterfulgt av mappennummeret. Til slutt skal det

lages en tom fil med navn **rett.txt** og denne filen skal ha rettigheter som er satt til en helt tilfeldig kombinasjon av x, r og w for alle de tre kategoriene. (hint til setting av rettigheter: ((**RANDOM % 8**)) gir et tilfeldig tall mellom 0 og 7)

Hvis mappen **ex** ikke finnes fra før, skal den lages. Hvis en undermappe (**ex500**, etc.) finnes fra før, skal den ikke lages men beholdes slik den er.

Skriv ditt svar her

1	
---	--

Maks poeng: 50

24 Prozess-Id

Med PowerShell kommandoer **ps** (alias for Get-Process) kan man liste Windows-prosesser:

```
PS C:\Users\os> ps
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
140	10	5992	10752	0,05	4568	0	audiodg
167	11	3284	14872	4,06	2300	1	conhost
236	11	1184	3408		376	0	csrss
230	12	1332	3832		464	1	csrss
123	8	1592	9368	0,06	3536	1	dllhost
337	35	23696	39108		792	1	dwm
1419	57	20004	63792	1,36	2568	1	explorer
0	0	0	4		0	0	Idle
901	21	4812	11292		552	0	lsass

Sett sammen en PowerShell-kommando (oneliner) som skriver ut alle prosesser som har prosess-Id lik 0.

Skriv ditt svar her

Maks poeng: 10

²⁵ Liste prosesser med høyest Id

Lag en PowerShell-kommando (oneliner) som lister de 10 prosessene i prosesslistingen som har høyest prosess-Id. Et eksempel på et resultat av en slik kommando er vist nedenfor:

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
908	58	45720	56276	0,34	4908	1	SearchUI
185	14	1724	7028		4076	0	svchost
533	49	17784	28152	0,73	3948	1	OneDrive
218	14	2920	10924	0,09	3916	1	MSASCuiL
450	26	10288	36644	0,75	3848	1	notepad
609	30	63632	78592	8,14	3688	1	powershell
123	8	1592	9368	0,06	3536	1	dllhost
125	9	1528	6772		3508	0	SppExtComObj
126	9	1948	8248		3200	0	sedsvc
780	36	20992	37100	0,48	2956	1	ShellExperience

Skriv ditt svar her

Maks poeng: 10