

Løsningsforslag konte-eksamen 2021 Operativsystemer

Prosesser

1)

Linux kommandolinje

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

Velg ett alternativ:

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



Riktig. 10 av 10 poeng. [Prøv igjen](#)

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.
- ☐ Fordele CPU-tid mellom prosesser.
- ☐ Håndtere internminnet (RAM) slik at prosesser ikke skriver over hverandres deler av minnet.
- ☐ Lage et filsystem som gir brukerne adgang til logiske filer.



Riktig. 10 av 10 poeng. [Prøv igjen](#)

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

Linux i praksis Oppgave 4-13 hadde med noen unntak individuelle svar. Se appendix. Mulig fremgangsmåte vist her.

4)

```
kan500@ex500:~$ cat file1
kyss 500
```

5) Man kan lete manuelt eller bruke en kommando som:

```
kan500@ex500:~$ find . -name file2
./fat/ipu3/dptf/file2
kan500@ex500:~$ cat ./fat/ipu3/dptf/file2
cat: ./fat/ipu3/dptf/file2: Permission denied
kan500@ex500:~$ chmod 600 ./fat/ipu3/dptf/file2
kan500@ex500:~$ cat ./fat/ipu3/dptf/file2
klin 500
```

6)

```
kan500@ex500:~$ sudo su
root@ex500:/home/kan500# cd
root@ex500:~# cat xfile
knin 500
```

7)

```
kan500@ex500:~$ groups
kan500 sudo
# Eller
kan500@ex500:~$ cat /etc/group | grep kan500
sudo:x:27:kan500
kan500:x:998:
```

8)

```

kan500@ex500:~$ sudo cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
kan500 ALL=(ALL) NOPASSWD:ALL

```

Det er linjen `%sudo ALL=(ALL:ALL) ALL` som gjør at alle vanlige brukere som er med i gruppen 'sudo' kan bli root med kommandoen `sudo su`. kan-brukeren må derfor være medlem av gruppen 'sudo' for å få til dette.

9) Det er linjen

```
kan500 ALL=(ALL) NOPASSWD:ALL
```

som gjør at kan-brukeren ikke trenger å taste eget passord for å kunne bli root.

10)

```

an500@ex500:~$ sudo grep kan500 /etc/shadow
kan500:$6$Vchk2wX07uFa.Lfa$LbTpKAn9Du8KiJeDWUqyAXIoowPA5q14h99CWF56X600gmvcF36SnYDohSTs/ftLUd885g9ZgarSPqxogyWBd/

```

Saltet er strengen mellom dollar-tegnene etter '6', for kan500: `Vchk2wX07uFa.Lfa`

11)

```

kan500@ex500:~$ echo aina7yaws | mkpasswd -m sha-512 -s -S Vchk2wX07uFa.Lfa
$6$Vchk2wX07uFa.Lfa$LbTpKAn9Du8KiJeDWUqyAXIoowPA5q14h99CWF56X600gmvcF36SnYDohSTs/ftLUd885g9ZgarSPqxogyWBd/

```

12)

```
oot@ex500:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
bc338c47fec1   ubuntu   "tail -f /dev/null"     5 days ago    Up 5 days                    jovial_jackson
```

container-ID: bc338c47fec1

Navn: jovial_jackson

13)

```
root@ex500:~# docker container exec -it bc338c47fec1 /bin/bash
root@bc338c47fec1:~# cat /root/xfile2
brum 500
```

eventuelt

```
root@ex500:~# docker exec -it bc cat /root/xfile2
brum 500
```

Assembly

14)

Assembly

Assemblykoden
mov \$0,%eax
gjør følgende:

Velg ett alternativ:

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

15)

Assembly

Assemblykoden

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

gjør følgende:

Velg ett alternativ:

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

Prosesser

16)

Prosesser

Hvilken Linux-kommando lister prosesser?

Velg ett alternativ:

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

17) 'cpu-migrations' angir hvor mange ganger prosessen har blitt flyttet fra en CPU til en annen og siden verdien er 0 betyr det at prosessen har kjørt på samme CPU hele tiden. 'context-switches' er antall ganger prosessen har vært involvert i en context-switch hvor den kjørende prosessen på CPUen skiftes ut. I dette tilfellet betyr det at prosessen har vært involvert i 52 context switches. 15,2 sekunder er hvor lang tid det tok å fullføre prosessen.

18) Scriptet starter de tre regn-prosessene som bakgrunnsjobber og alle tre vil derfor settes igang samtidig på det to CPUene, men bare to av dem kan da kjøre av gangen.

1. Alle de tre prosessene er involvert i ca 20 cpu-migrations fra en CPU til en annen og det betyr at Linux

schedulerer prosessene slik at de omtrent hvert sekund startes opp på en annen CPU enn de sist var. Alle prosessene kjører derfor ca ett sekund om gangen på en CPU, men på denne CPUen vil den hele tiden foreta context-switches med en annen regn-prosess og med andre prosesser. Det kunne vært slik at to prosesser alltid kjørte på samme CPU og den tredje hoppet frem og tilbake mellom de to, men slik er det ikke.

2. Når bare en enkelt regn-prosess kjører, vil den kun en gang iblant måtte context-switch med en annen prosess på samme CPU. De mest CPU-intensive av de andre jobbene vil da kjøres på den CPUen som regn ikke kjører på. Når tre regn-prosesser kjøres samtidig, vil det alltid være minst en regn-prosess på en CPU som ønsker å bruke CPUen og dermed hele tiden vil context-switch med de andre CPUene og andre regn-prosesser som er ready list på samme CPU. Spesielt på den CPUen som i øyeblikket har to regn-prosesser vil det bli mange context-switcher siden de to prosessene hele tiden bytter på å bruke CPUen.
3. En regn-prosess som kjører (tilnærmet) alene på en CPU bruker ca 15 sekunder. Tre slike prosesser trenger da totalt 45 CPU-sekunder for å fullføre og med to CPUer tilgjengelig vil de da bli ferdige på omtrent $45/2 = 22.5$ sekunder.

19)

1. Når scriptet startes med 'taskset -c 0' vil alle prosessen tvinges til å kjøre på CPU 0. Dermed vil ingen av regn-prosessene noen gang flyttes.
2. Nå vil det til enhver tid skje context-switcher mellom de tre regn-jobbene. Siden det i forrige eksempel alltid var en regn-prosess som kjørte alene på en CPU og da kun hadde noen få context-switches med de andre prosessene, vil det nå bli mange flere context-switches siden alle tre hele tiden ønsker å kjøre på samme CPU og dermed aldri får 'pauser' hvor de kjører nesten alene på en CPU (en prosess har som man ser av denne kjøringen ca 2000 context-switches når den hele tiden bytter med andre prosesser, mens i oppgave 18 kjørte de i 2/3 av tiden sammen med en annen prosess noe som dermed resulterte i omtrent 2/3 av 2000 context-switches.)
3. En regn-prosess som kjører (tilnærmet) alene på en CPU bruker ca 15 sekunder. Tre slike prosesser trenger da totalt 45 CPU-sekunder for å fullføre og med kun en CPU tilgjengelig vil de da bli ferdige på omtrent 45 sekunder.

20) Nå vil OS schedulere de to prosessene slik at de kjører på hver sin CPU og man forventer ingen cpu-migrations og at de begge bruker omtrent 15 sekunder på å fullføre, like lang tid som når en prosess kjørte alene. Man forventer litt fler context-switches enn når en prosess kjørte alene, siden de andre prosessene som kjører da får mer konkurranse om de to CPUene. En kjøring av perf2.sh ga følgende resultat:

```
Linux$ ./perf2.sh
```

```
Performance counter stats for './regn':
```

```

          0      cpu-migrations
        205      context-switches
```

```
15,036243672 seconds time elapsed
```

```
Performance counter stats for './regn':
```

```

          0      cpu-migrations
        209      context-switches
```

```
15,484064670 seconds time elapsed
```

Internminne

21)

Internminne

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

Velg ett alternativ

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

22)

Program-segmenter

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

Velg ett alternativ:

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

Riktig. 10 av 10 poeng. [Prøv igjen](#)

Bash-scripting

23)

```
#!/bin/bash

i=499

if [ ! -d ex ]
then
    mkdir ex
fi

for pw in $(cat pw)
do
```

```

((i++))

dir=ex/ex$i

if [ ! -d $dir ]
then
    mkdir $dir
    echo "$i $pw" > $dir/file1
    word=$(shuf -n1 ord.txt)
    echo "$word $i" > $dir/xfile

    touch $dir/rett.txt
    u=$(( RANDOM % 8 ))
    o=$(( RANDOM % 8 ))
    a=$(( RANDOM % 8 ))
    chmod $u$o$a $dir/rett.txt
fi

done

```

PowerShell

24)

```
Get-Process | Where-Object {$_.Id -eq 0}
```

alternativt

```
ps | ? {$_.id -eq 0}
```

25)

```
Get-Process | Sort-Object -des Id | Select-Object -first 10
```

alternativt

```
ps | sort -des Id | select -first 10
```

Appendix

Individuelle svar for oppgave 4-13, for hver oppgave:

```

4:
500: passord = aina7yaws, kyss 500 (intel2: yale 500)
501: passord = biteltrig, fald 501 (intel2: verp 501)
502: passord = dopplisoap, usse 502 (intel2: fyre 502)
503: passord = fiff7rife, goal 503 (intel2: aube 503)
504: passord = guru8pale, gess 504 (intel2: orke 504)
505: passord = iris3mast, koho 505 (intel2: gnav 505)
506: passord = klev3jobs, sina 506 (intel2: gods 506)
507: passord = lett1gram, huus 507 (intel2: boye 507)

```

```

508: passord = mora6exit, jams 508 (intel2: vald 508)
509: passord = otte6crop, waag 509 (intel2: ugge 509)
510: passord = reel1bind, pilt 510 (intel2: rime 510)
5:
500: klin 500 (intel2: hall 500)
501: pham 501 (intel2: krok 501)
502: baut 502 (intel2: vete 502)
503: rysj 503 (intel2: sokn 503)
504: vera 504 (intel2: anna 504)
505: gout 505 (intel2: flir 505)
506: ekse 506 (intel2: tind 506)
507: efor 507 (intel2: veel 507)
508: tapp 508 (intel2: regn 508)
509: lobb 509 (intel2: ofte 509)
510: tynn 510 (intel2: burt 510)
6:
500: knin 500 (intel2: amoy 500)
501: snue 501 (intel2: kuls 501)
502: kole 502 (intel2: rygh 502)
503: safe 503 (intel2: fert 503)
504: coen 504 (intel2: fryd 504)
505: vrid 505 (intel2: evne 505)
506: clou 506 (intel2: samd 506)
507: rock 507 (intel2: kiel 507)
508: nerd 508 (intel2: arno 508)
509: gang 509 (intel2: soss 509)
510: ting 510 (intel2: suni 510)
7:
500: kan500 sudo
501: kan501 sudo
502: kan502 sudo
503: kan503 sudo
504: kan504 sudo
505: kan505 sudo
506: kan506 sudo
507: kan507 sudo
508: kan508 sudo
509: kan509 sudo
510: kan510 sudo
10:
500: Vchk2wX07uFa.Lfa (intel2: q0yJzRdANQfPic1X)
501: /sgVix8p171JevZt (intel2: eFIgnDkNMpnRoS44)
502: oZq0ixHGob16FZpo (intel2: B0iy87gI13YLf8zI)
503: kRafHbs02zlnHGFK (intel2: vysA0N3k9c856N5b)
504: IDaT0kH8VbBOLyfs (intel2: ARnY6JdFB4vv01wz)
505: HBj0xJ7fhWMTZsBY (intel2: SKpMCQXLQx33Gozm)
506: AQoP6S0h2vWcKtk6 (intel2: Z9cD3gMuPh/0W1Nk)
507: vMu7r166zs28vRvt (intel2: 4vs6c9slc9JEhtIt)
508: ENUU1alvcy3Aw74Q (intel2: QyJmBnI03FA9PCc.)
509: IkiGL3BoAYGIvPUZ (intel2: qE8LDnls6vZ4r1/Q)
510: k0imwJCvZ1A4OMfB (intel2: siFbibS6lN3auj7s)
11:
500: echo "aina7yaws | mkpasswd -m sha-512 -s -S Vchk2wX07uFa.Lfa"
$6$Vchk2wX07uFa.Lfa$LbTpKAn9Du8KiJeDWUqyAXIoowPA5q14h99CWF56X600gmvcF36SnYDohSTs/ftLUd885g9ZgarSPqxogyWBd/
500: (intel2) echo "aina7yaws | mkpasswd -m sha-512 -s -S q0yJzRdANQfPic1X"
$6$q0yJzRdANQfPic1X$RkURkehpj50D36lIFqM/mH/9HH1WR8FwhXiJtZKFr3dsIFaoiLXqVBuWdi6rg0pcwEMHZAqwJE24kiOefvCYt0
501: echo "bite1trig | mkpasswd -m sha-512 -s -S /sgVix8p171JevZt"
$6$/sgVix8p171JevZt$LtlPNsWSeMueGNubYVbygteageyI5kL6RZjhI6V4fu7d.oxDl20d5hy01in2q3Xql6ZHbY3ZJ9RozMePzzINc/
501: (intel2) echo "bite1trig | mkpasswd -m sha-512 -s -S eFIgnDkNMpnRoS44"
$6$eFIgnDkNMpnRoS44$vPdCnWEs72D1bY2qHRg0gffgS.uuW8RudS0yNi6QRpNQIZMxF.JTE/2/IEnp10EvEDl1tAru8LsgKRNf83NRjf1

```

```

502: echo "dopp1soap | mkpasswd -m sha-512 -s -S oZq0ixHGob16FZpo"
$6$oZq0ixHGob16FZpo$mnqySVjgoPKytHhM1uJ98sq/1eNT4v33BhcZUFddHfKJEn3wu/eLpEIqoPhUYLvouk3IiiIAycdJNB61FdEzB/
502: (intel2) echo "dopp1soap | mkpasswd -m sha-512 -s -S B0iy87g1l3YLf8zI"
$6$B0iy87g1l3YLf8zI$.2unNczVbyhpNoLxkEz.J4S1TwLjhSd6dV0KF0UHUPECAHee0yqu0ByLxZ0cX3xmWAX9A7LaBvMDA3IMRGfp1
503: echo "fiff7rife | mkpasswd -m sha-512 -s -S kRafHbs02zlnHGFK"
$6$kRafHbs02zlnHGFK$InKXeoGgK5nNUIVHLZWRkSVx4Lo9Aejquz6VsOMPnOt2aU2Jtrc95UCwGkrSMehySMg/JQbzhCXwfbj0UeygQ.
503: (intel2) echo "fiff7rife | mkpasswd -m sha-512 -s -S vvsA0N3k9c856N5b"
$6$vvsA0N3k9c856N5b$1GhVRPoNyefBNDpog9wXm3SmvXBKdd5RMiua3GLvF3FBeC3sw6NeKPmn2UMt1/HB4/0QNG1Kh5FZCQ0tckn.80
504: echo "guru8pale | mkpasswd -m sha-512 -s -S IDaT0kH8VbBOLyfs"
$6$IDaT0kH8VbBOLyfs$WNRuKRrcLpbuc2FoKtdTFlyVqP3eKEC2xDXvqDPaQvW8rFQ6pWZgBzamkgkS1F08vH2Tj8eTiBEG5mb8BwCAf1
504: (intel2) echo "guru8pale | mkpasswd -m sha-512 -s -S ARnY6JdFB4vv01wz"
$6$ARnY6JdFB4vv01wz$dzx3TA4gF7Sp2YyTm0uIzjNItJ.06XJPFHnUyJdMdXMyrGEAkQZfweuOVdpEvmg6doNoW0Vr5q5uqx0XD0amH/
505: echo "iris3mast | mkpasswd -m sha-512 -s -S HBj0xJ7fhWMTZsBY"
$6$HBj0xJ7fhWMTZsBY$uThFgGyYVo6qvvcu6.MdFCGvX7E2vb6FNEa5fJzCGPBt6CNx/21SFBMRZ0JqbR9MM3qi7dx4RSp0mUXG8Ily31
505: (intel2) echo "iris3mast | mkpasswd -m sha-512 -s -S SKpMCQXLQx33Gozm"
$6$SKpMCQXLQx33Gozm$xaSSHj/M3/.uTi0iIM7E3ky/JRL1zbKCCoIkp6RiUz.opRgYsw6dcpr2Bm7Zp5DDwZnVllxGWo750i1HLUfvz.
506: echo "klev3jobs | mkpasswd -m sha-512 -s -S AqoP6S0h2vWcKtk6"
$6$AqoP6S0h2vWcKtk6$EtIYZrgpco.8vfwZpQkrGtEXJjivUwY4k6YGrvWvOJi1g46BiACCGDGeg2nDyNRfHiwtWvryJ618AVj2YFAT/
506: (intel2) echo "klev3jobs | mkpasswd -m sha-512 -s -S Z9cD3gMuPh/OWlNk"
$6$Z9cD3gMuPh/OWlNk$UnS8bSfKRmDvU.tACGooCkhehSdjQAdAHNax45zuyw36v2uJXsEck0eGgCs/jK34DZfWM3BPfcdSH4C8veXQv1
507: echo "lett1gram | mkpasswd -m sha-512 -s -S vMu7r166zs28vRvt"
$6$vMu7r166zs28vRvt$.rGhyA9wBj9bdyYlTdMA7wlaU8GYMJgPCPOyYM2U9sP29UwAt0yN0zlIgJUWcx3bw5410lnV7ESTWiQoZjX2F1
507: (intel2) echo "lett1gram | mkpasswd -m sha-512 -s -S 4vs6c9slc9JEhtIt"
$6$4vs6c9slc9JEhtIt$NflUUsmaWUPH2i7.2Jmw6qkSHBoZthTj6/L7PtaPFXnyuQZ/u8VK9zm4JfTJspKC.9A4xilX5zEKQa6bqElca.
508: echo "mora6exit | mkpasswd -m sha-512 -s -S ENUU1alvcy3Aw74Q"
$6$ENUU1alvcy3Aw74Q$Gfw9YR813j8pahLB8AJorJDR7zXkD419ebgrlNfE3Rh3ktgzfItFBCWBhMokTeEJGn0sVyZLqo3SC/KnlvegK.
508: (intel2) echo "mora6exit | mkpasswd -m sha-512 -s -S QyJmBnI03FA9PCc."
$6$QyJmBnI03FA9PCc.$UnXVroPdqcWwOnJd53brFPsnBGMz.UwNUKC/5MFYHleauZYkKU9H7ud3k8bn1RsdYUYzyT8gs2WUjUCSNPBN0
509: echo "otte6crop | mkpasswd -m sha-512 -s -S IkiGL3BoAYGIvPUZ"
$6$IkiGL3BoAYGIvPUZ$eAMUKH4gym6XcmU0q5fDnHR3WpYlqfwsOwx5S1UFH0tWJZPnyj9vRM6wixyv5Yuk0Br4i55KTziLj0F4SajN8/
509: (intel2) echo "otte6crop | mkpasswd -m sha-512 -s -S qE8LDnlS6vZ4r1/Q"
$6$qE8LDnlS6vZ4r1/Q$amj/M8kvLJVUEW9.v/TtXpM431y/y2ejjVyAuSPTHv5vapMPhVp4QKGP5ZVgxQBF7s.ZIEqfghRp0vJ7xbz0t.
510: echo "reel1bind | mkpasswd -m sha-512 -s -S k0imwJcVZ1A40MfB"
$6$k0imwJcVZ1A40MfB$XcFPakyZd6ALaM6w2/s.KFC.4rjycr5VXUHunw5SzgFk8T0C0rSFIqCmZL/nMiEY16tIm0/00Iw2BxYY1ZWN30
510: (intel2) echo "reel1bind | mkpasswd -m sha-512 -s -S siFbibS6lN3auj7s"
$6$siFbibS6lN3auj7s$SiF25WHxlgUjYTjZ9.B5eeF.KozUz1QbHkYgQx6e9bTjGQKJJtUTNrrhuMEFNqnkhOnsHoih7N47gCBZVRybtb1
12:

```

	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
500:	bc338c47fec1	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		jovial_jackson
501:	7f31e0b13403	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		focused_ride
502:	b34b7476d378	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		exciting_lumiere
503:	e18d81ef4995	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		gallant_brown
504:	97818020cd23	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		sad_chandrasekhar
505:	6413c55b24e8	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		practical_shirley
506:	15fdbb3a4d48	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		stupefied_perlman
507:	1b77e8c17d18	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		keen_keldysh
508:	b460f5fd28c9	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		exciting_dirac
509:	35ac4c9d490e	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		sweet_kilby
510:	52c2649fba3c	ubuntu	"tail -f /dev/null"	5 days ago	Up 5 days		upbeat_curie

13:

```

500: brum 500
501: ryan 501
502: broe 502
503: gynt 503
504: pren 504
505: olov 505
506: zebu 506
507: knin 507
508: rane 508
509: hamb 509
510: kvee 510

```

Individuelle svar for oppgave 4-13, for hver student:

Kandidat 500

```

4: password = aina7yaws
4: kyss 500 (intel2: yale 500)
5: klin 500 (intel2: hall 500)
6: knin 500 (intel2: amoy 500)
7: kan500 sudo
10: Vchk2wX07uFa.Lfa (intel2: q0yJzRdANQfPic1X)
11: echo "aina7yaws | mkpasswd -m sha-512 -s -S Vchk2wX07uFa.Lfa"
$6$Vchk2wX07uFa.Lfa$LbTpKAn9Du8KiJeDWUqyAXIoowPA5q14h99CWF56X600gmvcF36SnYDohSTs/ftLud885g9ZgarSPqxogyWBd/
11: (intel2) echo "aina7yaws | mkpasswd -m sha-512 -s -S q0yJzRdANQfPic1X"
$6$q0yJzRdANQfPic1X$RkURkehpj50D36lIFqM/mH/9HH1WR8FwhXiJtZKFr3dsIFaoiLXqVBuWdi6rg0pcwEMHZAqwJE24kiOefvCYt0
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
bc338c47fec1 ubuntu "tail -f /dev/null" 5 days ago Up 5 days jovial_jackson
13: brum 500

```

Kandidat 501

```

4: password = bite1trig
4: fald 501 (intel2: verp 501)
5: pham 501 (intel2: krok 501)
6: snue 501 (intel2: kuls 501)
7: kan501 sudo
10: /sgVix8p171JevZt (intel2: eFIgnDkNMpnRoS44)
11: echo "bite1trig | mkpasswd -m sha-512 -s -S /sgVix8p171JevZt"
$6$/sgVix8p171JevZt$LtlPNsWSeMueGNubYVbygteageyI5kL6RZjhI6V4fu7d.oxDl20d5hy01in2q3Xql6ZHbY3ZJ9RozMePxxINC/
11: (intel2) echo "bite1trig | mkpasswd -m sha-512 -s -S eFIgnDkNMpnRoS44"
$6$eFIgnDkNMpnRoS44$vPdCnWes72D1bY2qHRg0gffgS.uuW8RudS0yNi6QRpNQIZMxF.JTE/2/IEnp10EvEDltAru8LsgKRNf83NRjf1
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7f31e0b13403 ubuntu "tail -f /dev/null" 5 days ago Up 5 days focused_ride
13: ryan 501

```

Kandidat 502

```

4: password = dopp1soap
4: usse 502 (intel2: fyre 502)
5: baut 502 (intel2: vete 502)
6: kole 502 (intel2: rygh 502)
7: kan502 sudo
10: oZq0ixHGob16FZpo (intel2: B0iy87gIl3YLf8zI)
11: echo "dopp1soap | mkpasswd -m sha-512 -s -S oZq0ixHGob16FZpo"
$6$oZq0ixHGob16FZpo$mnqySVjgoPKytHhM1uJ98sq/1eNT4v33BhcZUFddHfKJEn3wu/eLpEIqoPhUYLvouk3IiiiAycdJNB61FdEzB/
11: (intel2) echo "dopp1soap | mkpasswd -m sha-512 -s -S B0iy87gIl3YLf8zI"
$6$B0iy87gIl3YLf8zI$.2unNczVbyhpNoLxkEz.J4S1TwLjhSd6dVOKFOUHuPECAHee0yqu0ByLxZ0cX3xmWAX9A7LaBvMDA3IMRGfp1
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b34b7476d378 ubuntu "tail -f /dev/null" 5 days ago Up 5 days exciting_lumiere
13: broe 502

```

```

Kandidat 503
4:  password = fiff7rife
4:  goal 503  (intel2: aube 503)
5:  rysj 503  (intel2: sokn 503)
6:  safe 503  (intel2: fert 503)
7:  kan503 sudo
10: kRafHbs02zlnHGFK (intel2: vysA0N3k9c856N5b)
11: echo "fiff7rife | mkpasswd -m sha-512 -s -S kRafHbs02zlnHGFK"
$6$kRafHbs02zlnHGFK$InKXeoGgK5nNUIVH1ZWRkSVx4Lo9Aejquz6VsOMPnOt2aU2Jtrc95UCwGkrSMehySMg/JQbzhCXwfbjOUeygQ.
11: (intel2) echo "fiff7rife | mkpasswd -m sha-512 -s -S vysA0N3k9c856N5b"
$6$vysA0N3k9c856N5b$1GhVRPoNyefBNDpog9wXm3SmvXBKdd5RMIuA3GLvF3FBec3sw6NeKPmn2UMt1/HB4/0QNG1Kh5FZCQOtckn.80
12: CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
e18d81ef4995    ubuntu    "tail -f /dev/null"    5 days ago    Up 5 days    gallant_brown
13: gynt 503

Kandidat 504
4:  password = guru8pale
4:  gess 504  (intel2: orke 504)
5:  vera 504  (intel2: anna 504)
6:  coen 504  (intel2: fryd 504)
7:  kan504 sudo
10: IDaT0kH8VbB0Lyfs (intel2: ARnY6JdFB4vv01wz)
11: echo "guru8pale | mkpasswd -m sha-512 -s -S IDaT0kH8VbB0Lyfs"
$6$IDaT0kH8VbB0Lyfs$WNRUKRrcLpbuc2FoKtDfTflyVqP3eKEC2xDXvqDPaQvW8rFQ6pWZgBzamkgkS1F08vH2Tj8eTIbEg5mb8BWcAF1
11: (intel2) echo "guru8pale | mkpasswd -m sha-512 -s -S ARnY6JdFB4vv01wz"
$6$ARnY6JdFB4vv01wz$dzx3TA4gF7Sp2YyTmOuIzjNItJ.06XJPFHnUyJdMdXMyrGEAkQZfweuOVdpEvmg6doNoW0Vr5q5uqX0xD0amH/
12: CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
97818020cd23    ubuntu    "tail -f /dev/null"    5 days ago    Up 5 days    sad_chandrasekhar
13: pren 504

Kandidat 505
4:  password = iris3mast
4:  koho 505  (intel2: gnav 505)
5:  gout 505  (intel2: flir 505)
6:  vrid 505  (intel2: evne 505)
7:  kan505 sudo
10: HBj0xJ7fhWMTZsBY (intel2: SKpMCQXLQx33Gozm)
11: echo "iris3mast | mkpasswd -m sha-512 -s -S HBj0xJ7fhWMTZsBY"
$6$HBj0xJ7fhWMTZsBY$uThFgGyYVo6qvvcu6.MdFCGvX7E2vb6FNEa5fJzCGPBt6CNx/21SFBMRZ0JQbR9MM3qI7dx4RSp0mUXG8I1Y31
11: (intel2) echo "iris3mast | mkpasswd -m sha-512 -s -S SKpMCQXLQx33Gozm"
$6$SKpMCQXLQx33Gozm$xaSSHj/M3/.uTi0iIM7E3ky/JRLizbKCcoIkp6RiUz.opRgYsw6dcpR2Bm7Zp5DDwZnVllxGWo750i1HLUfvz.
12: CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
6413c55b24e8    ubuntu    "tail -f /dev/null"    5 days ago    Up 5 days    practical_shirley
13: olov 505

Kandidat 506
4:  password = klev3jobs
4:  sina 506  (intel2: gods 506)
5:  ekse 506  (intel2: tind 506)
6:  clou 506  (intel2: samd 506)
7:  kan506 sudo
10: AQoP6S0h2vWcKtk6 (intel2: Z9cD3gMuPh/OWlNk)
11: echo "klev3jobs | mkpasswd -m sha-512 -s -S AQoP6S0h2vWcKtk6"
$6$AQoP6S0h2vWcKtk6$EtIYZrgpco.8vfwZpQKrGtEXJjivuYW4Ke6YGrvWvOJi1g46BiACCGDGeg2nDyNrfHiwtWvrYJ618AVj2YFAT/
11: (intel2) echo "klev3jobs | mkpasswd -m sha-512 -s -S Z9cD3gMuPh/OWlNk"
$6$Z9cD3gMuPh/OWlNk$UnS8bSfKRmDvU.tACGooCkhehSdjQAdAHNax45zuyw36v2uJXsEckOeGgCs/jK34DZfWM3BPfcdSH4C8veXQv1
12: CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
15fddb3a4d48    ubuntu    "tail -f /dev/null"    5 days ago    Up 5 days    stupefied_perlman
13: zebu 506

```

```
Kandidat 507
4:  password = lett1gram
4:  huus 507 (intel2: boye 507)
5:  efor 507 (intel2: veel 507)
6:  rock 507 (intel2: kiel 507)
7:  kan507 sudo
10: vMu7r166zs28vRvt (intel2: 4vs6c9slc9JEhtIt)
11: echo "lett1gram | mkpasswd -m sha-512 -s -S vMu7r166zs28vRvt"
$6$vMu7r166zs28vRvt$.rGhyA9wBj9bdyYlTdMA7wlaU8GYMJgPCP0yYM2U9sP29UwAt0yN0z1IgJUWcx3bw5410lnV7ESTWiQoZjX2F1
11: (intel2) echo "lett1gram | mkpasswd -m sha-512 -s -S 4vs6c9slc9JEhtIt"
$6$4vs6c9slc9JEhtIt$NflUasmaWUwPH2i7.2Jmw6qkSHBoZthTj6/L7PtaPFXnyuQZ/u8VK9zm4JfTJspKC.9A4xilX5zEKQa6bqElca.
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1b77e8c17d18 ubuntu "tail -f /dev/null" 5 days ago Up 5 days keen_keldysh
13: knin 507
```

```
Kandidat 508
4:  password = mora6exit
4:  jams 508 (intel2: vald 508)
5:  tapp 508 (intel2: regn 508)
6:  nerd 508 (intel2: arno 508)
7:  kan508 sudo
10: ENUU1alvcy3Aw74Q (intel2: QyJmBnIO3FA9PCc.)
11: echo "mora6exit | mkpasswd -m sha-512 -s -S ENUU1alvcy3Aw74Q"
$6$ENUU1alvcy3Aw74Q$Gfw9YR8l3j8pahLB8AJorjDR7zXkD4l9ebgrlNfE3Rh3ktgzfItFBCWBhMokTeEJGn0sVyZLqo3SC/KnlvegK.
11: (intel2) echo "mora6exit | mkpasswd -m sha-512 -s -S QyJmBnIO3FA9PCc."
$6$QyJmBnIO3FA9PCc.$UnXVroPdqcWWOnJd53brFPsnBGMz.UwNUKC/5MFYHieauZYkKU9H7ud3k8bn1RsdYUYzyT8gs2WUjUCSNPBN0
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b460f5fd28c9 ubuntu "tail -f /dev/null" 5 days ago Up 5 days exciting_dirac
13: rane 508
```

```
Kandidat 509
4:  password = otte6crop
4:  waag 509 (intel2: ugge 509)
5:  lobb 509 (intel2: ofte 509)
6:  gang 509 (intel2: soss 509)
7:  kan509 sudo
10: IkiGL3BoAYGIvPUZ (intel2: qE8LDnls6vZ4r1/Q)
11: echo "otte6crop | mkpasswd -m sha-512 -s -S IkiGL3BoAYGIvPUZ"
$6$IkiGL3BoAYGIvPUZ$eAMUKH4gym6XcmU0q5fDnHR3WpYlqfws0wx5S1UFH0tWJZPnyj9vRM6wixyv5Yuk0Br4i55KTziLjOF4SajN8/
11: (intel2) echo "otte6crop | mkpasswd -m sha-512 -s -S qE8LDnls6vZ4r1/Q"
$6$qE8LDnls6vZ4r1/Q$amj/M8kvLJVUEW9.v/TtXpM431y/y2ejjVyAuSPThv5vapMPHvP4QKGP5ZVgxQBF7s.ZIeqfghRp0vJ7xbzOt.
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
35ac4c9d490e ubuntu "tail -f /dev/null" 5 days ago Up 5 days sweet_kilby
13: homb 509
```

```
Kandidat 510
4:  password = reel1bind
4:  pilt 510 (intel2: rime 510)
5:  tynn 510 (intel2: burt 510)
6:  ting 510 (intel2: suni 510)
7:  kan510 sudo
10: k0imwJCvZ1A40MfB (intel2: siFbibS6lN3auj7s)
11: echo "reel1bind | mkpasswd -m sha-512 -s -S k0imwJCvZ1A40MfB"
$6$k0imwJCvZ1A40MfB$XcFPAkyZd6ALaM6w2/s.KFC.4rjycr5VXUHunw5SzgFk8TOC0rSFIqCMzL/nMiEYl6tIm0/00Iw2BxYY1ZWN30
11: (intel2) echo "reel1bind | mkpasswd -m sha-512 -s -S siFbibS6lN3auj7s"
$6$siFbibS6lN3auj7s$SiF25WHxlgUjYtJz9.B5eeF.KozUz1QbHkYgQx6e9bTjGQKJJtUTNrrhuMEFNqnkhOnsHoih7N47gCBZVRytb1
12: CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
52c2649fba3c ubuntu "tail -f /dev/null" 5 days ago Up 5 days upbeat_curie
13: kvee 510
```