OS OBLIG 2

- Jørgen Berntsen - s354410
- Atif Aziz - s354542
- Adrian August Grøtter - s354378
- Tony Strømsnæs - s354366

Uke 6

6.1.(Oblig) Les 5.2 'En linje høynivåkode kan gi flere linjer maskininstruksjoner' i Forelesningsnotatene, kompiler programmene, C-koden main og assembly-koden enlinje.s (den som starter med .globl enlinje), vis at en kjøring gir svaret 42 og forklar så hvorfor svaret blir 42 utifra assembly-koden.

```
gcc -no-pie enlinje.s main.c / commando som kompilerer Assembly koden enlinje og c
koden main

Vi har to variabler i assembly koden, svar(32) og memvar(10) under datafeltet som
legger variablene i ram

Først legger memvar i rbx
Også legges rbx til svar som ligger i ram = svar = memvar + svar
Også flyttes svar til rax som igjen returneres.
```

2.(Oblig) Bruk så det C-programet som gjør det samme som assembly-koden i forrige oppgave(deklarer int-variabler svar og memvar), kompiler det sammen med main C-programmet på samme måte og sjekk at svaret blir det samme. Kompiler C-funksjonen alene med gcc -S og studer den resulterende assembly-koden.

```
jorber@JBMain:~/uke6$ gcc main.c enlinje.c
jorber@JBMain:~/uke6$ ./a.out
Kaller enlinje()...
Svar = 42
```

Assembly:

```
enlinje:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    $32, -8(%rbp) #Flytter verdi 32 til register A
        movl    $10, -4(%rbp) #Flytter verdi 10 til register B
        movl    -4(%rbp), %eax #Flytter register B til RAM A
        addl    %eax, -8(%rbp) #Adderer RAM A til register A (10 + 32)
        movl    -8(%rbp), %eax #Flytter register A til RAM A (42)
        popq    %rbp            #Popper svaret for å returnere?
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
```

Hvor mange linjer leder den ene høynivå-koden til?

Den leder til 5 linjer assembly :

```
movl    $32, -8(%rbp) #Flytter verdi 32 til register A
movl    $10, -4(%rbp) #Flytter verdi 10 til register B
movl    -4(%rbp), %eax #Flytter register B til RAM A
addl    %eax, -8(%rbp) #Adderer RAM A til register A (10 + 32)
movl    -8(%rbp), %eax #Flytter register A til RAM A (42)
```

Hvordan endrer den resulterende assembly-koden seg om du bruker "long long" istedet for "int" for variablene i C-funksjonen?

```
jorber@JBMain:~/uke6/long$ diff enlinjeint.s enlinjelonglong.s
1c1
<       .file   "enlinjeint.c"
---
>       .file   "enlinjelonglong.c"
14,18c14,18
<       movl    $32, -8(%rbp)
<       movl    $10, -4(%rbp)
<       movl    -4(%rbp), %eax
<       addl    %eax, -8(%rbp)
<       movl    -8(%rbp), %eax
---
>       movq    $32, -16(%rbp)
>       movq    $10, -8(%rbp)
```

```
>       movq    -8(%rbp), %rax
>       addq    %rax, -16(%rbp)
>       movq    -16(%rbp), %ra
```

I long long versjonen ser det ut som størrelsen på registerene har doblet seg. Det gir mening siden int er 32 bit, og long long er 64 bit.

6.7.(Oblig) Lag en find-kommando på data2500 som lister alle filene i hjemmemappen din som ble endret i løpet av 30 januar 2022. (velg en annen dato å teste med hvis du ikke har endret noen filer denne dagen)

```
find ./ -newermt "2022-01-30" ! -newermt '2022-01-31'
```

6.8(Oblig) Lag en fil med navn fil.txt som inneholder

```
s802399@stud.hioa.no
s886878@stud.hioa.no
s886876@stud.hioa.no
s886885@stud.hioa.no
s886884@stud.hioa.no
s850806@stud.hioa.no
s886873@stud.hioa.no
s886888@stud.hioa.no
s808855@stud.hioa.no
s856627@stud.hioa.no
s886878@stud.hioa.no
s299507@stud.hioa.no
s850798@stud.hioa.no
s803434@stud.hioa.no
s886879@stud.hioa.no
s886877@stud.hioa.no
s959938@stud.hioa.no
s886880@stud.hioa.no
s826650@stud.hioa.no
s886882@stud.hioa.no
s886874@stud.hioa.no
s859987@stud.hioa.no
s803833@stud.hioa.no
s886885@stud.hioa.no
```

Lag så en kommando som fra denn filen lager en ny fil nyfil.txt hvor alle forekomster av stud.hioa.no er byttet ut med oslomet.no

MED SCRIPT

```
# /bin/bash
rm resultat.txt 2>/dev/null
while read LINE; do
    echo $LINE | sed s/stud.hioa.no/oslomet.no/ >> resultat.txt
done
cat resultat.txt
```

MED COMMAND

```
cat fil.txt | sed s/stud.hioa/oslomet/ >> nyfil.txt

jorber@JBMain:~/uke6/oppgave8$ ./script.sh < fil.txt
s802399@oslomet.no
s886878@oslomet.no
s886876@oslomet.no
s886885@oslomet.no
s886884@oslomet.no
s850806@oslomet.no
s886873@oslomet.no
s886888@oslomet.no
s808855@oslomet.no
s856627@oslomet.no
s886878@oslomet.no
s299507@oslomet.no
s850798@oslomet.no
s803434@oslomet.no
s886879@oslomet.no
s886877@oslomet.no
s959938@oslomet.no
s886880@oslomet.no
s826650@oslomet.no
s886882@oslomet.no
s886874@oslomet.no
s859987@oslomet.no
s803833@oslomet.no
s886885@oslomet.no
```

6.10.(Oblig) Lag en enlinjes Linux-kommando som skriver ut en liste med de 10 største filene i /etc, omtrent slik (på data2500):*

```
-rw-r--r-- 1 root root   70481 Jan 15  2021 mime.types
-rw-r--r-- 1 root root   26599 Jan 25 06:44 ld.so.cache
-rw-r--r-- 1 root root   12813 Mar 27  2021 services
-rw-r--r-- 1 root root   10593 Jan 30  2021 sensors3.conf
-rw-r--r-- 1 root root   10477 Feb  7  2020 login.defs
-rw-r--r-- 1 root root   10056 Dec  2  2020 nanorc
-rw-r--r-- 1 root root    9443 Jan  2 23:35 locale.gen
-rw-r--r-- 1 root root    6169 Feb 27  2021 sudo_logsrvd.conf
-rw-r--r-- 1 root root    5662 Dec 31 21:14 ca-certificates.conf
-rw-r--r-- 1 root root    5215 Feb 19  2021 manpath.config

ls -laS /etc | head -11
```

6.12. (Oblig) Lag et bash-script som skriver ut en oversikt tilsvarende den som er gitt nedenfor. All informasjon skal genereres dynamisk av scriptet og vil dermed avhenge av hvem som kjører det, hva scriptet heter, hvor det kjøres etc. Bruk scriptet fra forrige uke som teller filer og linker i dette scriptet (Hint: finn ut hva shell-variablene $0 og $$ er).

```
haugerud@data2500:~$ ./info.sh
Du har brukernavn haugerud og kjører scriptet ./info.sh med PID = 816133
på maskinen data2500 som kjører operativsystemet Linux
Oversikt over din hjemmekatalog /home/haugerud:
Filer: 95
Kataloger: 14
Linker: 0
Ditt default shell er /bin/bash og du befinner deg i
katalogen /home/haugerud
Totalt 32 brukere er oppført i passordfilen og det er definert 58 grupper
Oversikt over dine grupper:
haugerud : groups: cannot find name for group ID 101964
101964 sudo
```

```
#! /bin/bash

user=$(whoami)
script=$(basename $0)
os=$(uname)
pid=$(ps aux | grep $script | head -1 | awk '{print $2}')
directories=$(find ~/ -type d -not -path '*/.*' | wc -l)
files=$(find ~/ -type f -not -path '*/.*' | wc -l)
```

```
links=$(find ~/ -type l -not -path '*/.*' | wc -l)
currentDir=$(pwd)
usersInPasswd=$(cat /etc/passwd | wc -l)
groups=$(cat /etc/group | wc -l)

echo Du har brukernavn $user og kjører scriptet ./$script med PID = $pid
echo på maskinen $(hostname) som kjører operativsystemet $os
echo oversikt over din hjemmekatalog $HOME:
echo Filer: $files
echo Kataloger: $directories
echo Linker: $links
echo Ditt default shell er $SHELL og du befinner deg i
echo katalogen $currentDir
echo Totalt $usersInPasswd brukere er oppført i passordfilen og det er definert
$groups grupper
echo Oversikt over dine grupper:
echo $(groups $user)
```

6.16(Oblig) Dette er ukens s-server oppgave. Logg inn på din private intel2 s-server som du har brukt tidligere med noe slikt som

```
$ ssh -p 635 s135@intel2.vlab.cs.oslomet.no
```

Som før er portnummeret s-nummeret + 500, 635 er kun for s135. Finn din s-gruppe på Canvas, der ligger også passordet du trenger for å logge inn. På s-serveren er det fra før 1000 mapper hvor du skulle lete etter en fil i forrige uke. Nå er det lagt inn en fil en fil med navn dfil.txt i hver eneste av de 1000 mappene. Alle er like store og alle unntatt en inneholder teksten "Ikke denne". Filen du skal lete deg frem til skiller seg ut fra de andre ved at den sist ble endret 11. februar 2013 (2013-02-11 02:26:11). Å kunne finne en fil som ble endret på et gitt tidspunkt på en Linux-server kan i mange tilfeller være veldig nyttig og her må du prøve å få til det. Les ukens forelesningsnotater hvis du står fast. Innholdet i denne filen er som vanlig en 10-tegns kode som viser at du har løst oppgaven. For å sjekke at du har funnet rette streng, skriv den inn på siden https://nexus.cs.hioa.no/~haugerud/os6.php og du får beskjed om du har skrevet riktig. I tilegg blir du da med på konkurransen om å finne denne koden fortest mulig!

```
ls -ld */*/*/* */*/* */* * | grep dfil.txt | find -newermt 2013-2-11 ! -newermt
2013-2-12

s108 - LKFSK8HBkd
s15 - BIbFirWI9o
s253 - Nk5XPbjkq6
s194 - qZnRmBljcy
```

Uke 7

7.2.(Oblig) Logg inn på data2500 og last ned programmet regn med wget https://www.cs.hioa.no/~haugerud/regn til din egen hjemmekatalog (Bruk eventuelt wget http://data2500.ddns.net/regn hvis det ikke lastes ned). Programmet gir en feilmelding om ehco og det er meningen at det skal gjøre det, slik at error kan omdirigeres. Programmet regner på et problem i noen minutter (ca. et halvt på data2500 og Linux VM, ett minutt på s-server) og skriver tilslutt resultatet til skjermen. Start programmet slik at det blir en bakgrunnsprosess som skriver resultatet til filen res.txt når den er ferdig og som skriver feilmeldinger til err.txt. Hva skjer om du logger ut før regn er ferdig?

```
./regn >res.txt 2>err.txt &
```

Jobben kjøres fortsatt selv om jeg logger ut.

7.3(Oblig) Finn ut hvor mye CPU-tid regn-programmet bruker og hvor stor andel av den totale CPU-tiden det bruker ved å bruke kommandoen time foran selve kommandoen du kjører med $ time ./regn. For å få med prosentandel CPU, må man formatere output fra time, f. eks slik:

```
$ TIMEFORMAT="Real:%R User:%U System:%S %P%%"
```

der Real viser reell tid, User viser CPU-tid brukt i user-mode (prosessen selv som regner), System viser CPU-tid brukt i kernel-mode (av OS-kjernene) og %P gir prosentandel CPU denne prosessen har brukt. Hvis man legger denne kommandoen inn i ~/.bashrc vil dette formatet alltid brukes.

```
s354410@data2500:~/uke7/oppgave2$ time ./regn
./regn: line 6: ehco: command not found
./regn : regner....
./regn, resultat: 14045002650000
Real:25,364 User:23,999 System:1,359 99,97%
```

Prøv å kjøre programmet slik at også tidsbruken lagres i err.txt. Det siste er det ikke så enkelt å få til.....

```
s354410@data2500:~/uke7/oppgave2$ (time ./regn > res.txt) 2>err.txt
s354410@data2500:~/uke7/oppgave2$ cat err.txt
./regn: line 6: ehco: command not found
Real:24,457 User:23,189 System:1,264 99,98%
```

7.5(Oblig).Se igjen på top. Tast 1 mens top kjører. Hvor mange CPU'er har data2500? Hvor mange prosent CPU får hver prosess? Hvordan fordeles de to regne-jobbene på CPUer?

Begge får 100%, og de fordeles over begge to.

Start så fem prosesser i en for-løkke slik at de alle samtidig kjører ./a.out &. Hvordan fordeles CPUene mellom de 5 regne-jobbene? Tast f som gir mulighet til å se flere kolonner, flytt så pekeren ned til kolonnen "Last used CPU" med piltatsten og tast space og deretter q for å komme tilbake til top. Prøv å se hvilke CPUer dine 5 prosesser jobber på og prøv å forklare hvordan OS fordeler jobbene på CPU'ene.

De fordeles over alle CPU'ene. Det ser ut til at prosessene blir delt opp i veldig små deler, og CPU'en som er først tilgjengelig tar seg av neste bit.

7.9. På gruppens Linux-VM: lag en bruker for hvert medlem av gruppen, bruk samme brukernavn som dere har på data2500(s123456 eller lignende). Pass NØYE på å ha et ordentlig passord, minst 8 tegn og gjerne tall og spesialtegn, ellers vil dere kunne bli hacket. Prøv å logge deg på som en av brukerene du laget.

Hvilken kommando kan man bruke for å se hvilke grupper man tilhører?

```
groups
```

Prøv om din nye bruker kan bli root ved å kjøre kommandoen "sudo su". Forklar hvorfor det ikke går (HINT: se hvilke grupper din nye bruker er medlem av i forhold til den brukeren du fikk tildelt.)

```
Funker ikke fordi brukeren ikke er lagt til i "sudoers"-listen.
```

Se om du kan legge den nye brukeren din til riktig gruppe slik at den også kan bli root. Hva er det som gjør at medlemmene i denne gruppen får sudo-rettigheter?

I filen /etc/sudoers står det følgende:

```
# 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
```

Som sier at brukere som er medlev av sudo får alle rettigheter.

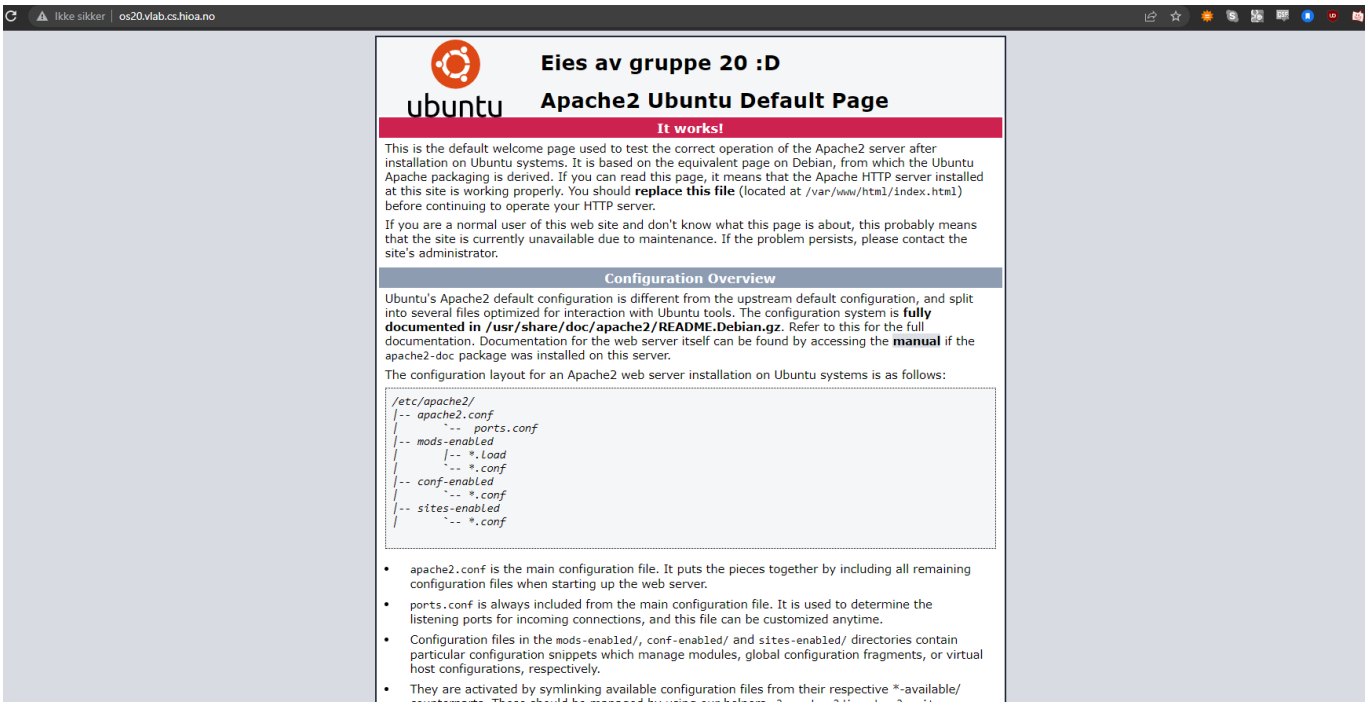
12.(Oblig) Installer en Apache webserver på gruppens Linux-VM med kommandoene

sudo apt update -y sudo apt install -y apache2

Deretter må du starte webserveren med

```
sudo service apache2 start
```

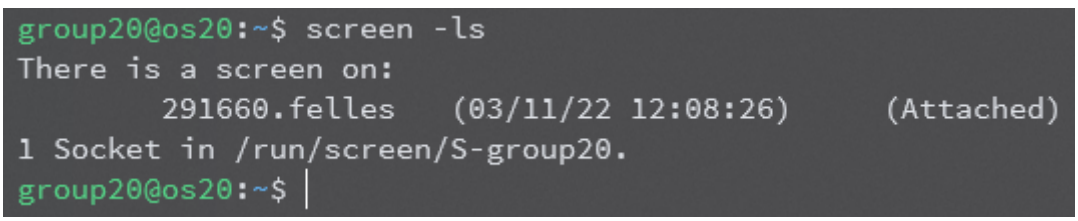
Test at Web-serveren virker ved å skrive inn http://os??.vlab.cs.hioa.no men der du bytter ut ?? med ditt gruppenummer. Alt du legger inn i katalogen /var/www/html kommer opp her. Altså har du nå din egen private webserver som kan nås fra overalt. Endre /var/www/html/index.html slik at man kan se hvilken gruppe som styrer denne webserveren.



**7.13 (Oblig) Installer først screen på Linux-VM om det ikke er installert:

```
sudo apt update -y
sudo apt install -y screen
```

Anta at du er på os-gruppe osXX hvor XX er ditt gruppenummer. Følg forelesningsnotatene, logg inn som brukeren groupXX og lag en screen session som kan deles av andre. Få så alle de andre på gruppen til å logge seg på med samme bruker, groupXX, og få deretter alle på gruppen til å koble seg til samme screen. Om du er alene på en gruppe, test med to forskjellige innloggingsvinduer. Dette kan være veldig praktisk om dere sitter rundt et bord og samarbeider om Linux-oppgaver og for eksempel skriver et script. Et av medlemmene kan da vise hvordan et problem løses, eventuelt kan alle prøve forskjellige kommandoer. Det gjelder bare å sørge for å være en av gangen, for alle kan taste samtidig.Det er også mulig å få til en delt screen mellom forskjellige brukere om man nøye følger punkt 4 på siden http://www.pc-freak.net/blog/share-screen-terminal-session-in-linux-screen-share-between-two-or-more-users-howto/.



7.14.(Oblig) Start fra data2500 og bruk ssh-copy-id til å sørge for at du kan logge inn til group-brukeren din på Linux-VM uten å taste passord (start fra s-server om du ikke kommer inn med ssh fra data2500 til Linux-VM). Se forelesningsnotater. Lag deretter et script som skal kjøres fra data2500. Scriptet skal først pakke ned mappen /tmp/dir på data2500 med tar cfz og lage en tgz-fil(pakket med tar og zippet med gzip, se notater). Deretter skal det flytte pakken til din bruker på Linux-VM med scp og pakke ut pakken i ~/tmp på Linux VM. Om denne mappen ikke finnes på Linux-VM, skal den lages først. Alle mellomlagrede filer (.tgz) på både data2500 og Linux VM skal slettes etterpå av scriptet. Når scriptet kjøres på data2500

```
s123456@data2500:~$ ./s.sh
dir.tgz
100%   43KB  12.0MB/s   00:00
```

skal det se omtrent slik ut på Linux-VM: group100@os100:~$ ls -l tmp/dir/ total 684 -rw-r--r-- 1 group100 group100 694186 Feb 16 18:08 auth.log -rw-r--r-- 1 group100 group100 4 Feb 16 18:08 enFil.txt**

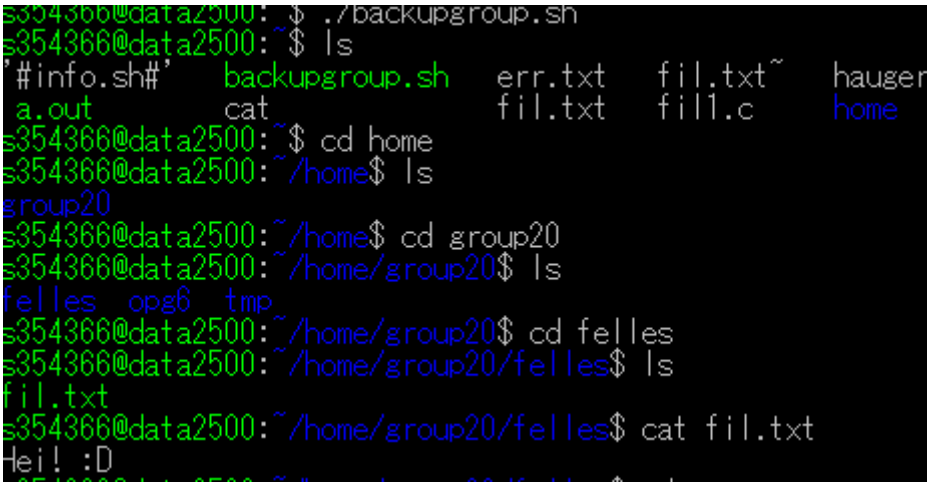
```
#! /bin/bash

tar cfz dir.tgz /tmp/dir
scp dir.tgz group20@os20.vlab.cs.hioa.no:~/
rm dir.tgz
ssh group20@os20.vlab.cs.hioa.no 'tar xfz dir.tgz'
ssh group20@os20.vlab.cs.hioa.no 'rm dir.tgz'
```

7.15(Oblig) (Bruk en s-server om du ikke kommer inn med ssh fra data2500 til Linux-VM) Installer først rsync på Linux-VM om den ikke er installert:

sudo apt update -y sudo apt install -y rsync

Lag et script som bruker rsync til å ta backup av av mappen /home/group?? med undermapper og filer på os??, hvor ?? er ditt gruppe-nummer og group?? er gruppens bruker på Linux-VM. La scriptet ta backup til en mappe ~/home for din bruker på data2500. Scriptet skal kjøres fra data2500, slik at du ikke gir andre med tilgang til Linux-VM tilgang til ditt område på data2500. NB! rsync må være installert på begge sider, så du må først installere rsync på Linux-VM. Hva skjer om du sletter en fil på Linux-VM og du etterpå gjør rsync? Bruk en rsync-kommandoen som gjør slik at filen da slettes der du lagrer backup(på data2500 i dette tilfellet). Bruk cron til å sørge for at det tas backup hver natt (kjør crontab -e på data2500) og få scriptet til å skrive en linje med

dato og klokkeslett for når backup ble tatt til en loggfil i din hjemme-mappe. Test ut med en hurtigere backup-frekvens (f.eks. ett minutt) for å sjekke at alt virker og sett så til nattlig backup. Pass på å bruke full path i crontab!




```
# Edit this file to introduce tasks to be run by cron.
#
    # Each task to run has to be defined through a single line
    # indicating with different fields when the task will be run
    # and what command to run for the task
    #
    # To define the time you can provide concrete values for
    # minute (m), hour (h), day of month (dom), month (mon),
    # and day of week (dow) or use '*' in these fields (for 'any').
    #
    # Notice that tasks will be started based on the cron's system
    # daemon's notion of time and timezones.
    #
    # Output of the crontab jobs (including errors) is sent through
    # email to the user the crontab file belongs to (unless redirected).
    #
    # For example, you can run a backup of all your user accounts
    # at 5 a.m every week with:
    # 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
    #
    # For more information see the manual pages of crontab(5) and cron(8)
    #
    # m h  dom mon dow   command

    * * * * * echo Utfører backup av filtreet. Dato: $(date)>>
/home/s354410/group20/backup/log.txt
    * * * * * rsync -a group20@os20.vlab.cs.hioa.no:/home/group20
/home/s354410/group20/backup
```

7.16 (Oblig) Dette er ukens s-server oppgave. Logg inn på din private intel2 s-server som du har brukt tidligere med noe slikt som

```
$ ssh -p 635 s135@intel2.vlab.cs.oslomet.no
```

Som før er portnummeret s-nummeret + 500, 635 er kun for s135. Finn din s-gruppe på Canvas, der ligger også passordet du trenger for å logge inn. På s-serveren er det fra før 1000 mapper hvor du skulle lete etter en fil i forrige uke. Nå er det lagt inn en tekst-fil i alle mapper. Du skal finne den av disse tekstfilene som er størst, inneholder flest antall bytes. Innholdet i denne filen (første linje) er som vanlig en 10-tegns kode som viser at du har løst oppgaven. For å sjekke at du har funnet rette streng, skriv den inn på siden https://nexus.cs.hioa.no/~haugerud/os7.php og du får beskjed om du har funnet riktig fil og streng. I tilegg blir du da med på konkurransen om å finne denne koden fortest mulig!

```
s108-asTSTa6FYy
s194 - dErdKJUzuW
s253 - IbMujzZQa5
s15 - g6NJwW9eu4
```

Uke 9

9.1.På forelesning ble output fra et par kommandoer brukt til å vise at kjernene på desktop'en rex var hyperthreading og til å vise hvilke to CPU'er mellom 0 og 7 (OS sin nummerering) som var thread-siblings, dvs deler ALU. Gjør det samme på Linux-VM. Er det thread-siblings på Linux-VM?

Ja, det er veldig mange.

```
group20@os20:~$ grep "" /sys/devices/system/cpu/cpu*/topology/thread_siblings_list
/sys/devices/system/cpu/cpu0/topology/thread_siblings_list:0,48
/sys/devices/system/cpu/cpu1/topology/thread_siblings_list:1,49
/sys/devices/system/cpu/cpu10/topology/thread_siblings_list:10,58
/sys/devices/system/cpu/cpu11/topology/thread_siblings_list:11,59
/sys/devices/system/cpu/cpu12/topology/thread_siblings_list:12,60
/sys/devices/system/cpu/cpu13/topology/thread_siblings_list:13,61
/sys/devices/system/cpu/cpu14/topology/thread_siblings_list:14,62
/sys/devices/system/cpu/cpu15/topology/thread_siblings_list:15,63
/sys/devices/system/cpu/cpu16/topology/thread_siblings_list:16,64
/sys/devices/system/cpu/cpu17/topology/thread_siblings_list:17,65
/sys/devices/system/cpu/cpu18/topology/thread_siblings_list:18,66
```

```
/sys/devices/system/cpu/cpu19/topology/thread_siblings_list:19,67
/sys/devices/system/cpu/cpu2/topology/thread_siblings_list:2,50
/sys/devices/system/cpu/cpu20/topology/thread_siblings_list:20,68
/sys/devices/system/cpu/cpu21/topology/thread_siblings_list:21,69
/sys/devices/system/cpu/cpu22/topology/thread_siblings_list:22,70
/sys/devices/system/cpu/cpu23/topology/thread_siblings_list:23,71
/sys/devices/system/cpu/cpu24/topology/thread_siblings_list:24,72
/sys/devices/system/cpu/cpu25/topology/thread_siblings_list:25,73
/sys/devices/system/cpu/cpu26/topology/thread_siblings_list:26,74
/sys/devices/system/cpu/cpu27/topology/thread_siblings_list:27,75
/sys/devices/system/cpu/cpu28/topology/thread_siblings_list:28,76
/sys/devices/system/cpu/cpu29/topology/thread_siblings_list:29,77
/sys/devices/system/cpu/cpu3/topology/thread_siblings_list:3,51
/sys/devices/system/cpu/cpu30/topology/thread_siblings_list:30,78
/sys/devices/system/cpu/cpu31/topology/thread_siblings_list:31,79
/sys/devices/system/cpu/cpu32/topology/thread_siblings_list:32,80
/sys/devices/system/cpu/cpu33/topology/thread_siblings_list:33,81
/sys/devices/system/cpu/cpu34/topology/thread_siblings_list:34,82
/sys/devices/system/cpu/cpu35/topology/thread_siblings_list:35,83
/sys/devices/system/cpu/cpu36/topology/thread_siblings_list:36,84
/sys/devices/system/cpu/cpu37/topology/thread_siblings_list:37,85
/sys/devices/system/cpu/cpu38/topology/thread_siblings_list:38,86
/sys/devices/system/cpu/cpu39/topology/thread_siblings_list:39,87
/sys/devices/system/cpu/cpu4/topology/thread_siblings_list:4,52
/sys/devices/system/cpu/cpu40/topology/thread_siblings_list:40,88
/sys/devices/system/cpu/cpu41/topology/thread_siblings_list:41,89
/sys/devices/system/cpu/cpu42/topology/thread_siblings_list:42,90
/sys/devices/system/cpu/cpu43/topology/thread_siblings_list:43,91
/sys/devices/system/cpu/cpu44/topology/thread_siblings_list:44,92
/sys/devices/system/cpu/cpu45/topology/thread_siblings_list:45,93
/sys/devices/system/cpu/cpu46/topology/thread_siblings_list:46,94
/sys/devices/system/cpu/cpu47/topology/thread_siblings_list:47,95
/sys/devices/system/cpu/cpu48/topology/thread_siblings_list:0,48
/sys/devices/system/cpu/cpu49/topology/thread_siblings_list:1,49
/sys/devices/system/cpu/cpu5/topology/thread_siblings_list:5,53
/sys/devices/system/cpu/cpu50/topology/thread_siblings_list:2,50
/sys/devices/system/cpu/cpu51/topology/thread_siblings_list:3,51
/sys/devices/system/cpu/cpu52/topology/thread_siblings_list:4,52
/sys/devices/system/cpu/cpu53/topology/thread_siblings_list:5,53
/sys/devices/system/cpu/cpu54/topology/thread_siblings_list:6,54
/sys/devices/system/cpu/cpu55/topology/thread_siblings_list:7,55
/sys/devices/system/cpu/cpu56/topology/thread_siblings_list:8,56
/sys/devices/system/cpu/cpu57/topology/thread_siblings_list:9,57
/sys/devices/system/cpu/cpu58/topology/thread_siblings_list:10,58
/sys/devices/system/cpu/cpu59/topology/thread_siblings_list:11,59
/sys/devices/system/cpu/cpu6/topology/thread_siblings_list:6,54
/sys/devices/system/cpu/cpu60/topology/thread_siblings_list:12,60
/sys/devices/system/cpu/cpu61/topology/thread_siblings_list:13,61
/sys/devices/system/cpu/cpu62/topology/thread_siblings_list:14,62
/sys/devices/system/cpu/cpu63/topology/thread_siblings_list:15,63
/sys/devices/system/cpu/cpu64/topology/thread_siblings_list:16,64
/sys/devices/system/cpu/cpu65/topology/thread_siblings_list:17,65
/sys/devices/system/cpu/cpu66/topology/thread_siblings_list:18,66
/sys/devices/system/cpu/cpu67/topology/thread_siblings_list:19,67
```

```
/sys/devices/system/cpu/cpu68/topology/thread_siblings_list:20,68
/sys/devices/system/cpu/cpu69/topology/thread_siblings_list:21,69
/sys/devices/system/cpu/cpu7/topology/thread_siblings_list:7,55
/sys/devices/system/cpu/cpu70/topology/thread_siblings_list:22,70
/sys/devices/system/cpu/cpu71/topology/thread_siblings_list:23,71
/sys/devices/system/cpu/cpu72/topology/thread_siblings_list:24,72
/sys/devices/system/cpu/cpu73/topology/thread_siblings_list:25,73
/sys/devices/system/cpu/cpu74/topology/thread_siblings_list:26,74
/sys/devices/system/cpu/cpu75/topology/thread_siblings_list:27,75
/sys/devices/system/cpu/cpu76/topology/thread_siblings_list:28,76
/sys/devices/system/cpu/cpu77/topology/thread_siblings_list:29,77
/sys/devices/system/cpu/cpu78/topology/thread_siblings_list:30,78
/sys/devices/system/cpu/cpu79/topology/thread_siblings_list:31,79
/sys/devices/system/cpu/cpu8/topology/thread_siblings_list:8,56
/sys/devices/system/cpu/cpu80/topology/thread_siblings_list:32,80
/sys/devices/system/cpu/cpu81/topology/thread_siblings_list:33,81
/sys/devices/system/cpu/cpu82/topology/thread_siblings_list:34,82
/sys/devices/system/cpu/cpu83/topology/thread_siblings_list:35,83
/sys/devices/system/cpu/cpu84/topology/thread_siblings_list:36,84
/sys/devices/system/cpu/cpu85/topology/thread_siblings_list:37,85
/sys/devices/system/cpu/cpu86/topology/thread_siblings_list:38,86
/sys/devices/system/cpu/cpu87/topology/thread_siblings_list:39,87
/sys/devices/system/cpu/cpu88/topology/thread_siblings_list:40,88
/sys/devices/system/cpu/cpu89/topology/thread_siblings_list:41,89
/sys/devices/system/cpu/cpu9/topology/thread_siblings_list:9,57
/sys/devices/system/cpu/cpu90/topology/thread_siblings_list:42,90
/sys/devices/system/cpu/cpu91/topology/thread_siblings_list:43,91
/sys/devices/system/cpu/cpu92/topology/thread_siblings_list:44,92
/sys/devices/system/cpu/cpu93/topology/thread_siblings_list:45,93
/sys/devices/system/cpu/cpu94/topology/thread_siblings_list:46,94
/sys/devices/system/cpu/cpu95/topology/thread_siblings_list:47,95
```

9.2. Kompiler det RAM-intensive programmet mem.c demonstrert på forelesning

```
group20@os20:~$ for i in 0 1; do time ./a.out& done
[1] 20593
[2] 20594
group20@os20:~$
Real:10.015 User:9.998 System:0.000 99.82%
Real:10.030 User:10.011 System:0.000 99.81%
```

```
group20@os20:~$ for i in 0 1; do time taskset -c 0 ./a.out& done
[1] 20597
[2] 20598
group20@os20:~$
```

```
Real:20.047 User:10.020 System:0.000 49.98%
Real:20.052 User:10.020 System:0.000 49.97%
```

```
Det tar dobbelt så lang tid når jeg velger hvilken CPU de skal kjøre på.
```

9.3.Ta utgangspunkt i scriptet regn som ble brukt i forelesningen til å finne ut om CPUene på desktopen rex bruker hyperthreading.

```
#! /bin/bash

(( max = 3000000 ))
(( i = 0  ))
(( sum = 0  ))

while (($i < $max))
do
        (( i += 1 ))
    (( sum += i  ))
done
#echo $0, resultat: $sum
```

Sørg først for at du kan kjøre dette og ta tiden på det som i de forrige oppgavene. På forelesningen ble kjøring og tidtaging av flere instanser av dette programmet til å avgjøre om Linux-VM har hyperthreading cores, på desktopen rex, der konklusjonen ble at det var fire hyperthreading kjerner (cores), som OS betraktet som totalt 8 CPUer. Men denne metoden er det vanskelig å bruke på Linux-VM fordi man der kun får tilgang til 2 CPUer av serverens 48 cores. Din VM er egentlig en docker-container som er startet med docker container run --cpus="2" slik at den bare får det som tilsvarer 2 CPU-er med CPU-tid. I oppgave 1 denne uken, fant du ut hvilke CPUer som er siblings. Prøv å finne ut av effekten av hyperthreading for CPUene på Linux-VM ved å samtidig kjøre regn-scriptet ved hjelp av taskset på to CPU-siblings og på to CPUer som ikke er siblings og ta tiden på dem. Utgjør dette noen forskjell utifra tiden det tar? Hva kan du konkludere utifra dette?

```
group20@os20:~$ for i in 0 48; do time taskset -c $i ./regn.sh& done
[1] 21016
[2] 21017
group20@os20:~$
Real:22.098 User:21.941 System:0.004 99.30%
Real:22.111 User:21.947 System:0.004 99.27%
```

```
for i in 0 48; do time taskset -c $i ./regn.sh& done^C
[1]-  Done                    time taskset -c $i ./regn.sh
[2]+  Done                    time taskset -c $i ./regn.sh
group20@os20:~$ for i in 0 1; do time taskset -c $i ./regn.sh& done
[1] 21033
[2] 21034
group20@os20:~$
Real:12.990 User:12.979 System:0.000 99.92%
Real:13.310 User:13.287 System:0.000 99.83%
```

Det er stor forskjell mellom å bruke hyperthreading og to forskjellige CPU-kjerner. Det er fordi det er bare en ALU per CPU, og i et CPU-intensivt program som "regn" hvor alt foregår på ALU'en, så må de to prosessene stå i kø for å bruke denne i hyperthreading, mens de slipper det med hver sin ALU og CPU-kjerne.

9.5. Docker LAB oppgaver

1. Hello World

```
root@os20:/home/group20# docker container run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

2.Running Image

```
root@os20:/home/group20# docker image pull alpine
Using default tag: latest
latest: Pulling from library/alpine
59bf1c3509f3: Pull complete
Digest: sha256:21a3deaa0d32a8057914f36584b5288d2e5ecc984380bc0118285c70fa8c9300
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

```
root@os20:/home/group20# docker container run alpine ls -l
total 56
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 bin
drwxr-xr-x    5 root     root           340 Mar  2 14:23 dev
drwxr-xr-x    1 root     root          4096 Mar  2 14:23 etc
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 home
drwxr-xr-x    7 root     root          4096 Nov 24 09:20 lib
drwxr-xr-x    5 root     root          4096 Nov 24 09:20 media
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 mnt
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 opt
dr-xr-xr-x 2511 root     root             0 Mar  2 14:23 proc
drwx------    2 root     root          4096 Nov 24 09:20 root
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 run
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 sbin
drwxr-xr-x    2 root     root          4096 Nov 24 09:20 srv
dr-xr-xr-x   13 nobody   nobody           0 Mar  2 14:23 sys
drwxrwxrwt    2 root     root          4096 Nov 24 09:20 tmp
drwxr-xr-x    7 root     root          4096 Nov 24 09:20 usr
drwxr-xr-x   12 root     root          4096 Nov 24 09:20 var
```

```
root@os20:/home/group20#  docker container run alpine echo "hello from alpine"
hello from alpine
```

```
root@os20:/home/group20# docker container run -it alpine /bin/sh
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin
srv    sys    tmp    usr    var
```

```
/ # exit
root@os20:/home/group20# docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@os20:/home/group20# docker container ls -a
CONTAINER ID   IMAGE         COMMAND                  CREATED              STATUS
PORTS     NAMES
9c2a08af1a43   alpine        "/bin/sh"                About a minute ago   Exited
(130) 21 seconds ago              recursing_hertz
```

```
63d31428d089   alpine        "echo 'hello from al…"   3 minutes ago        Exited
(0) 3 minutes ago                flamboyant_darwin
b125142270ef   alpine        "ls -l"                  4 minutes ago        Exited
(0) 4 minutes ago                xenodochial_ellis
4e3e578d0664   hello-world   "/hello"                 10 minutes ago       Exited
(0) 10 minutes ago               naughty_lehmann
bb20e94a493b   hello-world   "/hello"                 About an hour ago    Exited
(0) About an hour ago            unruffled_mcnulty
507e90be950c   hello-world   "/hello"                 About an hour ago    Exited
(0) About an hour ago            confident_archimedes
```

For å navngi en container "test"

```
root@os20:/home/group20# docker container run -it --name test alpine /bin/sh
```

3.Deletion

```
$ rm -rf /
/ # ls
/bin/sh: ls: not found
/ # ls
/bin/sh: ls: not found
/ # cd
/bin/sh: cd: can't cd to /root: No such file or directory
/ # ls -la
/bin/sh: ls: not found
```

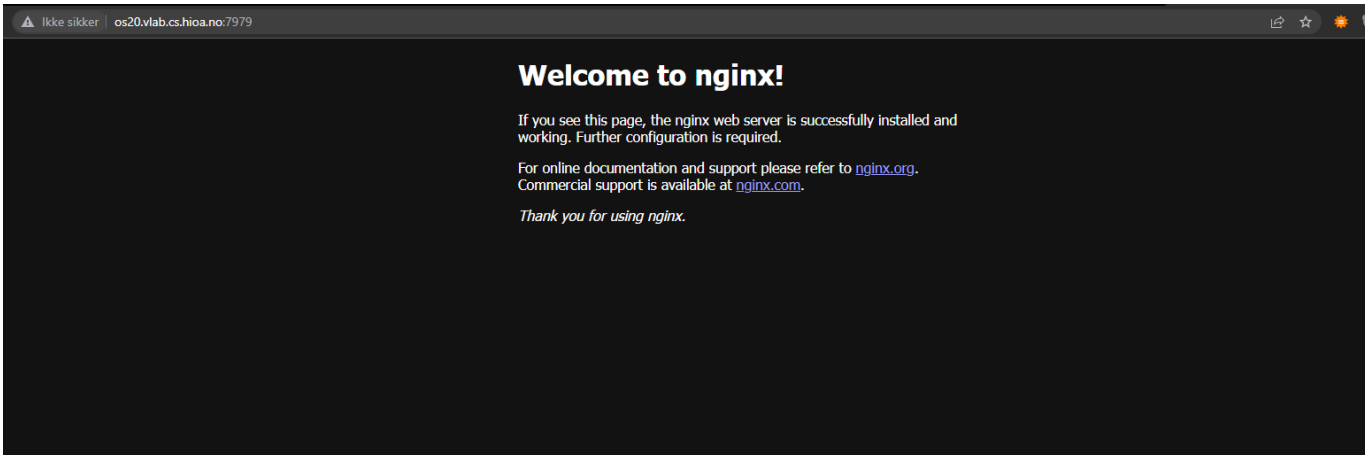
```
root@os20:/home/group20# docker container ls -as
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS
PORTS     NAMES                 SIZE
4887da041a22   alpine        "/bin/sh"                26 seconds ago   Exited (0)
7 seconds ago                peaceful_hamilton      8B (virtual 5.58MB)
eb1b453294cd   alpine        "/bin/sh"                5 minutes ago    Exited
(127) 2 minutes ago              exciting_aryabhata     0B (virtual 5.58MB)
```

```
root@os20:/home/group20# docker container rm naughty_lehmann
naughty_lehmann

root@os20:/home/group20# docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
PORTS     NAMES
4887da041a22   alpine    "/bin/sh"                4 minutes ago    Exited (0) 4
```

```
minutes ago               peaceful_hamilton
eb1b453294cd   alpine    "/bin/sh"                9 minutes ago    Exited (127) 7
minutes ago              exciting_aryabhata
282c722abe3f   alpine    "/bin/sh"                11 minutes ago   Exited (127) 9
minutes ago              pedantic_chaplygin
36b804dd4fff   alpine    "/bin/sh"                12 minutes ago   Exited (0) 12
minutes ago               test
a74b05a229e8   alpine    "/bin/sh --name test"    13 minutes ago   Exited (2) 13
minutes ago               nostalgic_agnesi
9c2a08af1a43   alpine    "/bin/sh"                16 minutes ago   Exited (130) 15
minutes ago              recursing_hertz
63d31428d089   alpine    "echo 'hello from al…"   18 minutes ago   Exited (0) 18
minutes ago               flamboyant_darwin
b125142270ef   alpine    "ls -l"                  19 minutes ago   Exited (0) 19
minutes ago               xenodochial_ellis
```

```
root@os20:/home/group20# docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        latest    c059bfaa849c   3 months ago   5.58MB
hello-world   latest    feb5d9fea6a5   5 months ago   13.3kB
root@os20:/home/group20# docker image rm feb5d9fea6a5
Untagged: hello-world:latest
Untagged: hello-
world@sha256:97a379f4f88575512824f3b352bc03cd75e239179eea0fecc38e597b2209f49a
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
```

4.A basic webserver




```
Kjører i bakgrunn
```

```
root@os20:/home/group20# docker container run -p 7979:80 -d nginx
c48e742e40da9208b4b8dd6bd6465649533ea29db60923714e79fad1bd453775
```

5. Executing

```
root@os20:/home/group20# docker container exec -it
c48e742e40da9208b4b8dd6bd6465649533ea29db60923714e79fad1bd453775 bash
root@c48e742e40da:/#
```

9.6.Start et ubuntu-container og inkluder -p 8081:80, slik at man kan koble til en webserver på port 8081 på Linux-VM. Gå inn i containeren på kommandolinjen og installer apache2 webserver og få den til å kjøre. Prøv også å få webserveren til å starte og å endre innholdet på webserveren fra kommandolinjen på Linux-VM. Sjekk at du kan se webserveren på

```
http://os100.vlab.cs.hioa.no:8081/
```

hvor du må bytte ut os100 med din Linux-VM. Når du har fått installert apache2 og fått den til å kjøre, gå ut av containeren med CTRL-p CTRL-q (da fortsetter den å kjøre i bakgrunnen) og list kjørende containere med

```
# docker container ps -a
```

Da vil du for containeren du har installert apache i, få opp en linje som ser ut omtrent slik:

```
3376df05f434        ubuntu        "/bin/bash"         3 minutes ago       Up 3
minutes                 0.0.0.0:8081->80/tcp   elegant_almeida
```

Lag så en kopi av denne containeren med

```
docker container commit 3376df05f434 apacheubuntu
```

eventuelt ved å bruke docker export og import. Etter å ha gjort dette, kan du starte opp en ny container med apache installert ved

```
# docker container run -it -d -p 8082:80 apacheubuntu /bin/bash
```

Deretter kan du starte apache i denne containeren ved å gjøre

```
docker container exec IDTILNYCONTAINER /bin/bash -c "/etc/init.d/apache2 start"
```

hvor IDTILNYCONTAINER er id til den nye containeren du nettopp startet (returneres av kommandoen du startet med, kan også sees med # docker container ps -a). Endre så /var/www/html/index.html på begge docker-instansene slik at du kan se hvem som er hvem når du åpner webserverene i en browser.

```
root@os20:/home/group20# docker container run -it -d -p 8082:80 apacheubuntu
/bin/bash
a82a72a317dd5a40b8efd44bf054eb0e9681466b841b2047ed87a030348d9cc6
root@os20:/home/group20# docker container ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
PORTS                  NAMES
a82a72a317dd   apacheubuntu   "/bin/bash"              27 seconds ago   Up 26
seconds               0.0.0.0:8082->80/tcp   awesome_chebyshev
f33a256aec01   ubuntu         "/bin/bash"              10 minutes ago   Up 10
minutes               0.0.0.0:8081->80/tcp   loving_satoshi
843e75d2b66c   ubuntu         "-p 8081:80 /bin/bash"   10 minutes ago   Created
zealous_hermann
94a83aab0150   ubuntu         "-it -p 8081:80 /bin…"   10 minutes ago   Created
charming_aryabhata
c48e742e40da   nginx          "/docker-entrypoint.…"   20 minutes ago   Up 20
minutes               0.0.0.0:7979->80/tcp   jovial_austin
fbcc326b26b4   nginx          "/docker-entrypoint.…"   23 minutes ago   Exited (0)
21 minutes ago                          tender_williamson
d35bcf4d428e   nginx          "/docker-entrypoint.…"   24 minutes ago   Exited (0)
23 minutes ago                          confident_williamson
294347c6baea   nginx          "/docker-entrypoint.…"   25 minutes ago   Exited (0)
24 minutes ago                          xenodochial_bassi
root@os20:/home/group20# docker container exec a82a72a317dd /bin/bash -c
"/etc/init.d/apache2 start"
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 172.17.0.4. Set the 'ServerName' directive globally to suppress this
message
 *
root@os20:/home/group20# docker container exec a82a72a317dd /bin/bash -c "echo
container 1 > /var/www/html/index.html"
```

```
root@os20:/home/group20# docker container exec f33a256aec01 /bin/bash -c "echo
container 2 > /var/www/html/index.html"
```

(Oblig)9.9 Dette er ukens s-server oppgave. Logg inn på din private intel2 s-server som du har brukt tidligere med noe slikt som

```
$ ssh -p 635 s135@intel2.vlab.cs.oslomet.no
```

Som før er portnummeret s-nummeret + 500, 635 er kun for s135. Finn din s-gruppe på Canvas, der ligger også passordet du trenger for å logge inn. På s-serveren står nå en container og kjører. Kjøre en docker-kommando som gjør at du får tilgang til et root-shell på denne containeren. Finn deretter en system-fil som inneholder informasjon om hvilken Ubuntu-versjon som kjører her. På slutten av denne filen er det lagt til en 10-tegns kode med tilfeldige tegn som viser at du har løst oppgaven. For å sjekke at du har funnet rette streng, skriv den inn på siden https://nexus.cs.hioa.no/~haugerud/os9.php og du får beskjed om du har funnet riktig fil og streng. I tilegg blir du da med på konkurransen om å finne denne koden fortest mulig!

```
s108 - nIXYMfQts3
s253 - Wcz3qJ1Knu
s15 - csnu5N2wTH
s194 - Har levert på nettsiden men får ikke til å kjøre docker container nå.
```