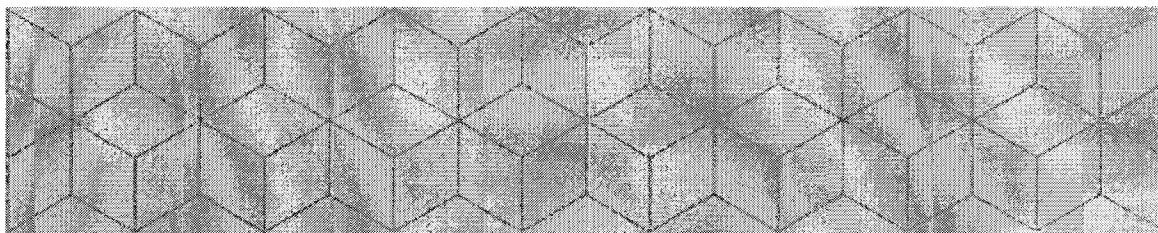


EKSAMEN

Emnekode: ITF22515	Emne: Operativsystemer med Linux
Dato: 7. desember 2016	Eksamenstid: 09:00 – 13:00
Hjelpemidler: Alle trykte og skrevne	Faglærere: Jan Høiberg, Tore Petter Engen
<p>Om eksamensoppgavene:</p> <p>Oppgavesettet består av 7 sider, inkludert denne forsiden. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene.</p> <p>Eksamenen har 5 oppgaver med i alt 20 deloppgaver. Noen av deloppgavene har underpunkter. Les hver oppgave nøye før du begynner på besvarelsen.</p> <p>Alle Linux-kommandoer og shellprogrammer skal skrives i Bourne Again Shell. Legg vekt på å skrive en kortfattet og lett forståelig besvarelse med ryddig kode.</p>	
<p>Sensurfrist: 4. januar 2017</p> <p>Karakterene er tilgjengelige for studenter på studentweb senest 2 virkedager etter oppgitt sensurfrist. www.hiof.no/studentweb</p>	



Oppgave 1: Generelt om operativsystemer og programmer (20%)

- a) Nedenfor er det gitt en liste med 16 ord/begreper som alle er navn eller betegnelser på deler av datasystemer:

emacs, Windows 10, C, Linux, CPU, http, sed, internminne, Java, Unix, TCP/IP, sftp, OS X, Powerpoint, MS-DOS, solid-state drive

Del inn disse ordene i følgende fem kategorier (skriv dem opp i fem kolonner med kategorien øverst):

1. Operativsystem
2. Maskinvare
3. Applikasjon
4. Protokoll
5. Programmeringsspråk

- b) I programmeringsspråket C kan man allokere (tildele) minne dynamisk, mens et program kjører. C har ingen «garbage-collector» som «gjenvinner» minne som ikke lenger er i bruk, slik som f.eks. i Java. Det er i stedet programmereren selv som har ansvar for å frigjøre dynamisk allokert minne som ikke lenger er i bruk.

Hvis et C-program ikke frigjør dynamisk allokert minne, kan vi få «minnelekkasje». Vi kan da risikere at programmet etterhvert «spiser opp» nesten alt av primærminne som er tilgjengelig på maskinen.

Forklar *kort* hvorfor dette vil kunne føre til at systemet etterhvert går veldig sakte og at disken samtidig brukes intenst.

- c) Du skal lage et program som inneholder en del løkker og mange tunge regneoperasjoner. Programmet skrives først i programmeringsspråket *S1*, men det viser seg å gå altfor sakte. Du skriver deretter samme program i språket *S2* og det går nå 100 ganger så fort. Du er likevel ikke fornøyd og skriver det til slutt i språket *S3*, hvor det går enda 10 ganger fortere.

Det tre språkene som du har brukt er Java, C og bash.

- Hvilket av de tre programmeringsspråkene er *S1*, *S2* og *S3*?
- Forklar *kort* hva de store forskjellene i hurtighet kommer av.

(Slutt på oppgave 1)

Oppgave 2: Linux-kommandoer og -verktøy (20%)

a) Skriv Linux-kommandoer som utfører følgende:

1. Legger til en linje som inneholder den absolutte søkestien til stående katalog, på slutten av filen `kataloger.txt`.
2. Åpner filen `prog.c` i `emacs` som en bakgrunnsprosess.
3. Gir deg selv skrive- og leserettigheter og alle andre brukere kun leserettigheter til filen `foto.jpg` i stående katalog.
4. Lagrer resultatet av kommandoen `date` i variabelen `today`.
5. Skriver ut innholdet av variabelen `today`.
6. Skriver ut alle linjer i filen `/etc/passwd` som inneholder strengen `root`.

b) Du har en tekstfil `readme.txt` og utfører kommandoene:

```
cat readme.txt > /dev/null
cat readme.txt
```

Forklar kort forskjellen på de to kommandoene og hva de vil resultere i.

(Oppgave 2 fortsetter på neste side)

c) Her er et utdrag av manualsiden for kommandoen `date`:

```
NAME
    date - print or set the system date and time
SYNOPSIS
    date [OPTION]... [+FORMAT]
DESCRIPTION
    Display the current time in the given FORMAT

    -d STRING
        display time described by STRING, not 'now'

    -r FILE
        display the last modification time of FILE

    FORMAT controls the output.  Interpreted sequences are:

    %A    full weekday name (e.g., Sunday)
    %B    full month name (e.g., January)
    %c    date and time (e.g., Thu Mar  3 23:05:25 2005)
    %d    day of month (e.g., 01)
    %m    month (01..12)
    %F    full date; same as %Y-%m-%d
    %R    24-hour hour and minute (hh:mm)
    %T    24-hour hour, minute and second (hh:mm:ss)
    %Y    year (e.g., 1970)
```

Eksempler på output fra `date`:

```
$ date
Tue 22. Nov 14:12:28 +0100 2016

$ date +%A-%B-%d-%Y
Tuesday-November-22-2016
```

En fil `data.txt` har følgende metadata:

```
$ ls -l data.txt
-rw-r--r-- 1 janh janh 13 Oct  6 16:07 data.txt
```

Skriv en kommando som bare skriver ut tidspunktet for forrige gang innholdet av filen `data.txt` ble endret. Output fra kommandoen `skal` skrives ut på denne formen:

```
2016-10-06_16:07:00
```

(Slutt på oppgave 2)

Oppgave 3: Shellprogrammering (25%)

a) Hva utføres av shellprogrammet nedenfor?

```
#!/bin/bash
antall=0
for filnavn in *
do
    if [ -x "$filnavn" ]
    then
        ((antall++))
    fi
done
echo $antall
```

b) I denne oppgaven skal du lage et shellprogram for å gjøre backup av en katalog:

- Programmet skal kopiere en hel katalog, med alle filer og underkataloger.
- Den kopierte katalogen skal legges som en underkatalog til katalogen ~/backup.
- Den kopierte katalogen skal få et *unikt* navn som entydig viser hvilken katalog det er tatt backup av og tidspunktet for backup.
- Hvis programmet kjøres uten argumenter, skal det kopiere stående katalog.
- Hvis programmet blir kjørt med ett argument skal dette tolkes som et katalognavn og backup skal tas av denne katalogen. Hvis katalogen ikke eksisterer skal programmet avsluttes med en passende feilmelding. Bruker av programmet kan angi katalogen med både absolutt og relativ søkesti.
- Hvis det gis to eller flere argumenter, skal programmet avsluttes med en feilmelding.

Hint:

- Kommandoen `cp -r` kan brukes til å kopiere et helt katalogtre.
- For å lage et unikt katalognavn kan man f.eks. ta utgangspunkt i det opprinnelige katalognavnet og bruke kommandoen `date` til legge til en streng basert på nåværende tidspunkt. Se oppgave 2 c) for eksempler på bruk av `date`.

(Slutt på oppgave 3)

Oppgave 4: Regulæruttrykk (10%)

I denne oppgaven skal du løse et svært lite «kryssord» med regulæruttrykk.

Eksempel:

Det er gitt fire regulæruttrykk:

A^*	Matcher 0 eller flere A
$(B C)^*$	Matcher 0 eller flere B og/eller C
AB	Matcher bare AB
$[CA]^*$	Matcher 0 eller flere C og/eller A

Vi setter nå disse uttrykkene inn i et 2x2 «kryssord» som skal fylles ut med fire bokstaver, slik at hvert av de fire regulæruttrykkene matches i hver sin vannrette rad eller loddrette kolonne. Her er et slikt «kryssord» med løsning:

	AB	$[CA]^*$
A^*	A	A
$(B C)^*$	B	C

Oppgave:

Fyll inn fire bokstaver i «kryssordet» nedenfor, slik at de to bokstavene som står etter hvert av de to vannrette og loddrette regulæruttrykkene matcher regulæruttrykket.

	$[^SPEAK]^+$	$EP IP EF$
$HE LL O^+$		
$[PLEASE]^+$		

(Slutt på oppgave 4)

Oppgave 5: Servere og nettverk (25%)

- a) Du administrerer webserveren til din bedrift, og ser i loggene at mange fremdeles etterspør mappen `/bilder_2014` (som ble slettet i 2015). Du ønsker i stedet at de skal nå mappen `/bilder_2016`. Hvordan gjør du dette, og hva slags statuskode ønsker du at apache skal svare med når brukerne forsøker å slå opp mappa `/bilder_2014`?
- b) Du vil sette opp to VirtualHosts på din webserver. Det er viktig at e-post til serveradmin blir sendt til forskjellige steder, og det er viktig at hvert nettsted har forskjellige logger. Hvordan kan en slik VirtualHost konfigurasjon se ut?
- c) Serveren går veldig tregt, og du mistenker at det er noen som prøver å "bryte seg inn" på din webserver. Hvordan kan du sjekke dette litt nærmere?
- d) Du ønsker å lage en mailliste som heter `alle_studenter@skolen.no`. Enkelte av studentene har eksterne mailadresser, og noen er interne. Hvordan kan du gjøre dette når du administrerer en Linuxserver med Sendmail installert?
- e) Forklar hvordan "Spamassassin" jobber i forhold til Sendmail og hvordan den sjekker ut hva som er "spam" og hva som ikke er "spam".
- f) Hva er forskjellen på protokollene "TCP" og "UDP" i forhold til IPTABLES?
- g) Forklar kort forskjellen på "Input-kjedet" og "Forward-kjedet" i IPTABLES.
- h) Forklar hva mappen `/etc/skel` gjør, og nevnt noen gode eksempler på bruk.
- i) Hvordan ville du gått frem for å opprette 1500 nye brukere i et Linuxsystem?
- j) Hva er "runlevels" på et Linuxsystem?
- k) Hvorfor er det viktig å partisjonere diskene i et Linuxsystem?

(Slutt på oppgave 5)