

EKSAMEN

Emnekode: ITF22515	Emne: Operativsystemer med Linux
Dato: 24. november 2015	Eksamenstid: 09:00 – 13:00
Hjelpemidler: Alle trykte og skrevne	Faglærere: Jan Høiberg, Tore Petter Engen
<p>Om eksamensoppgavene:</p> <p>Oppgavesettet består av 8 sider, inkludert denne forsiden. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene.</p> <p>Eksamenen har 5 oppgaver med i alt 27 deloppgaver. Les hver oppgave nøye før du begynner på besvarelsen.</p> <p>Alle Linux-kommandoer og shellprogrammer skal skrives i Bourne Again Shell. C-kode skal være ANSI C. Legg vekt på å skrive en kortfattet og lett forståelig besvarelse med ryddig kode.</p>	
<p>Sensurdato: 15. desember 2015</p> <p>Karakterene er tilgjengelige for studenter på studentweb senest 2 virkedager etter oppgitt sensurfrist. Følg instruksjoner gitt på: www.hiof.no/studentweb</p>	

Oppgave 1: Multitasking, prosesser og tråder (15%)

- a) Forklar *kort* hvordan et operativsystem på en datamaskin med én enkel CPU (med bare én kjerne) tilsynelatende klarer å kjøre flere prosesser samtidig.
- b) Et meget CPU-intensivt C-program som brukes til værvarsling, kan i løpet av en halv time (30 minutter) regne ut værprognosen for neste dag på et Linux-system som har én enkel CPU. Programmet skal kjøres fire (4) ganger med fire forskjellige sett input-data. De fire kjøringene av programmet startes samtidig. Omtrent hvor lang tid vil det ta før alle de fire værprognosene er ferdige? Begrunn svaret.
- c) Programmet for værvarsling skrives om slik at det bruker fire tråder. Hver tråd regner ut en værprognose basert på hvert sitt sett med input-data. Omtrent hvor lang tid vil det ta før de fire værprognosene er ferdige, når de kjøres på samme Linux-system som ovenfor? Begrunn svaret.
- d) Tråd-programmet for værvarsling flyttes til et annet Linux-system som har samme hardware som ovenfor, men som har *fire* identiske enkle CPU'er. Omtrent hvor lang tid bør det nå ta før de fire værprognosene er ferdige? Begrunn svaret.
- e) Meteorologene som kjører værvarslingsprogrammet ønsker å se fortløpende hvor stor andel av CPU(ene) og primærminnet som programmet bruker. Hvordan kan de enkelt gjøre dette på et Linux-system?

(Shutt på på oppgave 1)

Oppgave 2: Linux kommandoer og verktøy (20%)

a) Skriv Linux-kommandoer som utfører følgende:

1. Sletter filen `/tmp/log`.
2. Gir deg som eier alle rettigheter til filen `bare_min.txt`, mens ingen andre brukere får noen rettigheter.
3. Skriver ut statusinformasjon bare om de prosessene på maskinen der prosessinformasjonen inneholder strengen `emacs`.

b) Hva utføres av denne kommandolinjen?:

```
maskin=$(hostname); os=$(uname);
```

c) Følgende Linux-kommandoer blir utført:

```
echo "Kanye West" > data.txt
echo "Cashmere Cat" > data.txt
cat data.txt | grep Kanye > ny.txt
```

Hva inneholder filen `ny.txt` etter at kommandoene er blitt kjørt? Begrunn svaret.

d) Hva blir resultatet av at følgende kommandoer blir kjørt? Begrunn svaret.

```
mkdir top_secret
chmod 000 top_secret
cp /etc/passwd top_secret
```

e) Filen `/proc/meminfo` inneholder oppdatert informasjon om Linux-systemets minnebruk. Her er et eksempel på hvordan et utsnitt av denne filen kan se ut:

```
MemTotal:      8039276 kB
MemFree:       3906840 kB
Buffers:       410156 kB
Cached:        2312828 kB
SwapCached:    0 kB
Active:        2308000 kB
Inactive:      1403820 kB
Unevictable:   32 kB
Mlocked:      32 kB
SwapTotal:    8249340 kB
SwapFree:     8249340 kB
```

Bruk verktøyene `grep`, `tr` og `sed` til å skrive en Linux-kommando på bare én linje, som henter verdiene for `MemTotal` og `MemFree` fra `/proc/meminfo` og skriver ut:

```
Totalt minne: 8039276 kB
Ledig minne: 3906840 kB
```

(Slutt på oppgave 2)

Oppgave 3: Shellprogrammering (25%)

- a) Hva utføres av shellprogrammet nedenfor?

```
#!/bin/bash
cd $1
for f in `ls *.mp3`
do
    mv $f ~/musikkarkiv/
done
```

- b) Skriv om shellprogrammet i deloppgave a), slik at det blir mer robust for feil i input.
- c) Brukerne på et Linux-system er vanligvis definert i "passordfilen" /etc/passwd . Starten på denne filen kan se slik ut:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
ftp:x:99:99:Anonymous FTP:/local/it/ftp:/bin/sync
snort:x:10004:1005:Snort IDS:/var/log/snort:/bin/false
nobody4:x:65534:65534:Nobody:/:/bin/sync
mroot:x:0:0:Tcsh Root account:/local/it:/bin/bash
studwww:x:24:24:web server daemon:/:/bin/bash
snort:x:10044:1005:Snort IDS:/var/log/snort:/bin/false
snort:x:10088:1005:Snort IDS:/var/log/snort:/bin/false
janh:x:1000:1000:Jan Høiberg:/home/janh:/bin/bash
```

Feltene på en linje er skilt med tegnet ':' (kolon). Det første feltet på hver linje inneholder brukernavn, det tredje feltet inneholder numerisk brukerid (UID).

Skriv et shellprogram som leser /etc/passwd og utfører følgende syv operasjoner:

1. Skriver ut hvilken maskin (host) scriptet kjøres på og hvilket OS som kjøres.
2. Gir feilmelding og avslutter hvis passordfilen ikke finnes eller ikke kan leses.
3. Skriver ut en lang listing av passordfilens egenskaper (se eksemplet under).
4. Skriver ut antall brukere definert i passordfilen.
5. Skriver ut brukernavn for alle brukere som har UID lik 0 (null).
6. Skriver ut alle brukernavn som forekommer flere ganger.
7. Skriver ut alle UID som forekommer flere ganger.

(Oppgave 3 fortsetter på neste side)

For filen gitt ovenfor kan programmet gi følgende utskrift:

```
Dette scriptet kjører på ask, operativsystemet er Linux
-rw-r--r-- 1 root root 732 Nov 20 15:02 /etc/passwd
Antall brukere: 12
root har uid = 0
mroot har uid = 0
Brukernavn som forekommer flere ganger:
    3 snort
UID som forekommer flere ganger:
    2 0
    2 65534
```

Noen hint som kan brukes til å løse oppgaven:

- Deloppgave 2 b) ovenfor inneholder noen nyttige kommandoer...
- Operasjon nr 5 som programmet skal utføre løses enklest ved å pipe innholdet av passordfilen til en løkke som leser feltene på filen linje for linje.
- Et utsnitt av manualsiden for kommandoen `uniq` ser slik ut:

NAME

`uniq` - report or omit repeated lines

SYNOPSIS

`uniq` [OPTION]... [INPUT [OUTPUT]]

DESCRIPTION

Filter adjacent matching lines from INPUT (or standard input), writing to OUTPUT (or standard output). With no options, matching lines are merged to the first occurrence.

`-c` prefix lines by the number of occurrences
`-d` only print duplicate lines

(Slutt på oppgave 3)

Oppgave 4: C-programmering (15%)

Følgende C-funksjon er gitt:

```
void min_max(int A[], int n, int min, int max)
{
    /*
     * Finner minste og største verdi i arrayen A
     * som har lengde n. Minste verdi returneres i
     * min, største verdi returneres i max.
     */

    int i;
    min = max = A[0];

    for (i = 1; i < n; i++)
    {
        if (A[i] < min)
            min = A[i];
        else if (A[i] > max)
            max = A[i];
    }
}
```

- Denne funksjonen vil ikke fungere slik som beskrevet i kommentarene i koden. Hvorfor?
- Skriv en ny versjon av funksjonen `min_max` som fungerer korrekt.
- Skriv et hovedprogram som fyller en array med tilfeldige tall med verdier større enn 0 og mindre eller lik 100, og deretter tester om funksjonen `min_max` virker. Programmet skal skrive ut både innholdet av arrayen og resultatet fra `min_max`.

(Slutt på oppgave 4)

Oppgave 5: Servere og nettverk (25%)

- a) Hva står `DocumentRoot` for i Apache-sammenheng?
- b) Du vil passordbeskytte et webområde kalt `hemmelig` (mappen `hemmelig` ligger i `DocumentRoot`). Hvordan kan du gjøre dette med Apache-konfigurering?
- c) Du vil sperre et bestemt subnett ute fra en bestemt katalogstruktur på webserveren. Hvordan gjør du det?
- d) Du ser følgende linje i Apache-konfigurasjonen din, hva gjør den?
`Redirect permanent /studentbedrift http://www.studentbedrift.no`
- e) Beskriv kort hva de forskjellige linjene i følgende direktiv gjør:
- ```
<VirtualHost *:80>
ServerAdmin webmaster@studentbedrift.no
DocumentRoot /var/www/studentbedrift
ServerName www.studentbedrift.no
ErrorLog logs/studentbedrift.no -error_log
CustomLog logs/studentbedrift.no -access_log common
</VirtualHost>
```
- f) Vi prater ofte om "default policy" når vi snakker om iptables. Hva menes med dette? Forklar også hva "accept" og "drop" betyr i forhold til dette.
- g) Hva gjør følgende to linjer i et iptables-skript? Forklar parameterne/variablene.
- ```
IPT -A FORWARD -p tcp -d $INT_NET --dport 22 -j ACCEPT
$IPT -t nat -A PREROUTING -d $EXT_IP -p tcp --dport 22 -j DNAT --to
192.168.1.3:22
```
- h) Du ønsker at alle nye brukere på ditt Linuxsystem skal få VI som standard teksteditor. Du vil også at de skal få en standard nettside med teksten "Jeg har dessverre ikke oppdatert min nettside..." på adressen `www.studentbedrift.no/~brukernavn`. Hvordan vil du gjøre dette? Anta at Apache er satt opp til å tillate at brukere kan publisere html-filer i katalogen `html-docs` på sitt hjemmeområde.
- i) Du skal slette en bruker manuelt fra systemet ditt. Hvilke filer trenger du å oppdatere for at all informasjon om brukeren skal bli borte?

- j) Du ønsker å lage en mailliste som heter `medlemmer@studentbedrift.no`. Enkelte av mottagerne har eksterne mailadresser, og noen er interne. Hvordan kan du gjøre dette når du er systemansvarlig for Linuxserveren som administrerer mailserveren `studentbedrift.no` (anta den kjører Sendmail som mailserver).
- k) Forklar hvordan Spamassassin jobber i forhold til Sendmail, og hvordan den sjekker ut hva som er "spam" og hva som ikke er "spam".

(Slutt på oppgave 5)