

Kapittel 14

X Window System

Dette kapitlet er hentet fra
Praktisk Linux, 3. utgave av Geir Maribu
Mer info på tisip.no/boker/plx

Læringsmål:

Etter å ha lest dette kapitlet skal du ha lært

- om forholdet mellom X Window System og de grafiske brukergrensesnittene
- om vindusbehandler
- om forskjellen mellom vindusbehandler og skrivebordsbehandler
- å starte Linux med eller uten X Window System
- å tilpasse konfigurasjonsfiler for X Window System

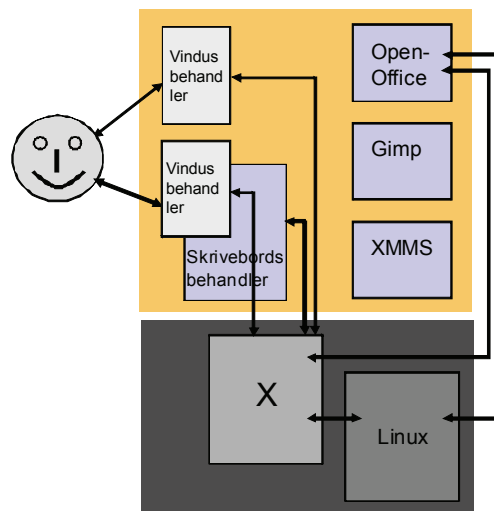
14.1 Linux og grafiske brukergrensesnitt

Basis for alle grafiske brukergrensesnitt i Linux er X Window System. Systemet så dagens lys første gang midt på 80-tallet og var da et tillegg til operativsystemet Unix for dem som ønsket å kjøre grafiske brukergrensesnitt. Du treffer X Window System under flere navn: X-Window, X11, X11R6 eller bare X.

Det finnes flere implementasjoner av X. Den mest brukte i Linux-distribusjoner per i dag er X.Org. Det er fri programvare slik dette er definert av Free Software Foundation. Tidligere ble XFree86 mye brukt.

Når vi nedenfor snakker om X, så gjelder det en vilkårlig X-implementasjon, for eksempel X.Org.

Figuren nedenfor viser sammenhengen mellom Linux og de forskjellige grafiske systemene som kjører på en Linux-maskin.



Figur 14.1 Linux og grafisk brukergrensesnitt

Til venstre i figuren er brukeren, som sitter foran en Linux-maskin og jobber via mus/tastatur og en skjerm med grafisk brukergrensesnitt. Du vil ofte se forkortelsen GUI brukt som en fellesbetegnelse for alt som har med grafiske brukergrensesnitt å gjøre. GUI står for Graphical User Interface.

Brukeren er i kontakt med en skrivebordsbehandler eller en vindusbehandler på skjermen. Brukeren styrer interaksjonen mellom GUI og bruker via mus/tastatur og via det som observeres på skjermen.

En skrivebordsbehandler er et integrert og komplett GUI bestående av mange komponenter, for eksempel skrivebordsflate med ikoner, panellinje nederst på skjer-

men, vindusbehandler etc. Eksempler på skrivebordsbehandlere er Gnome og KDE.

En vindusbehandler er et GUI bestående kun av denne ene komponenten, nemlig vindusbehandleren, som tar seg av virkemåten til vinduene og interaksjonen med GUI via mus/tastatur, det vil si hvordan man lager nye vinduer, flytter vinduer, lukker vinduer, endrer størrelse på vinduer etc. Eksempler på vindusbehandlere er BlackBox, Windowmaker, fvwm, twm, Sawfish, kwin, Metacity pluss mange flere. Metacity er standard vindusbehandler for Gnome, mens kwin er standard vindusbehandler for KDE.

La oss bruke et eksempel i den videre forklaringen: Brukeren klikker med musa inne i vinduet til brukerprogrammet for å gjøre et valg. Det som da skjer, er at vindusbehandleren (som kontrollerer og styrer vinduene) mottar denne hendelsen som en melding om et museklikk. Denne meldingen sendes til X og deretter videre til Linux, som overrekker museklikkmeldingen til det aktuelle brukerprogrammet, der museklikket blir tolket og det utføres en handling på grunnlag av museklikket. Handlingen kan for eksempel være å merke et avsnitt med tekst. Neste trinn er at brukerprogrammet sender en melding tilbake for å vise fram den aktuelle tekstblokken som merket.

I eksemplet ovenfor ble museklikket utført inne i vinduet til brukerprogrammet. Dersom museklikket i stedet hadde blitt utført på selve vindusrammen, for eksempel for å flytte vinduet eller endre størrelse på vinduet, hadde det vært en oppgave kun for vindusbehandleren, og det hadde ikke vært nødvendig å sende museklikkmeldingen hele veien fram til brukerprogrammet.

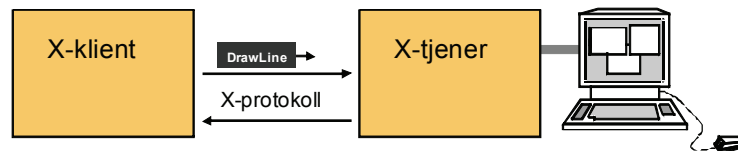
Lukking av vinduet, derimot, fører til at vindussystemet lukker det aktuelle vinduet. Men i tillegg sendes det et signal til Linux om å avslutte programmet som kjørte i det aktuelle vinduet.

Legg også merke til at det ikke er et hierarkisk forhold mellom vindusbehandler eller skrivebordsbehandler, det vil si GUI, og selve brukerprogrammet. Alle disse opererer som separate programmer i Linux, og de fungerer sammen fordi det sendes meldinger mellom GUI og brukerprogrammet. Disse meldingene sendes via X og Linux.

14.2 Hva er X?

X tilbyr grunnleggende funksjonalitet til de grafiske brukergrensesnittene i Linux. Det er X som tilbyr funksjoner for å endre størrelse på vinduer, lage vinduer, lukke vinduer, flytte vinduer etc. samt for interaksjon via mus og tastatur. Vindusbehandleren, som er et eget program atskilt fra X, tilbyr et brukergrensesnitt mot denne funksjonaliteten slik at disse vindusoperasjonene kan utføres med tastaturet eller med musa.

X er basert på et klient–tjener-system. En klient i denne sammenhengen er et brukerprogram, for eksempel programmet `xclock` eller `xterm`, som står i kontakt med en X-tjener (eng.: X Server) slik som vist i figuren nedenfor.



Figur 14.2 Forholdet mellom x-klient og x-tjener

Tjeneren styrer/kontrollerer skjermen, det vil si tegner vinduer, tekst, linjer, bilder, sirkler etc. etter pålegg fra brukerprogrammene, altså X-klientene. En X-tjener kan håndtere tegning på skjermen for flere programmer samtidig – hvert program vil typisk vise utdataene i hvert sitt vindu. X-tjeneren tar seg også av tastetrykk og museklikk. X-tjeneren som styrer mus/tastatur og skjerm, sender tastetrykk og museklikk til aktuell X-klient, alt etter hvilket vindu det er klikket i.

X-klienten er et program under X som kommuniserer med X-tjeneren. En X-klient kan være en tekstbehandler, et regneark eller de typiske X-programmene `xclock`, `xterm`, `xload`, `xedit` etc.

En X-klient som ønsker å tegne/skrive på en skjerm, sender en bestemt melding til X-tjeneren. Denne meldingen beskriver i detalj hva som skal utføres. I figur 14.2 ser vi at X-klienten er i ferd med å sende over en melding om tegning av en linje på skjermen. Meldingene sendes over mellom X-klient og X-tjener i et definert meldingsformat. Dette meldingsformatet kalles X-protokoll. Én melding kan for eksempel tegne en linje, en annen en sirkel og en tredje skrive ut en tekst.

X-tjeneren har flere oppgaver: Den tillater at flere X-klienter har tilgang til samme skjermen, den tolker meldingene som kommer fra X-klientene, den sender inndata tilbake til klientene som meldinger, den utfører grafiske operasjoner på skjermen, og den inneholder kompliserte datastrukturer for å kunne håndtere vinduer, musemerker, skrifttyper (fonter) og farger lokalt på X-tjeneren. Det betyr at trafikken mellom X-klient og X-tjener reduseres til et minimum, da X-klienten kun trenger å si fra om hva som skal utføres på skjermen; den trenger ikke å spesifisere i detalj hvordan ting skal utføres, og den trenger heller ikke å sende med detaljerte spesifikasjoner av skrifttyper og farger. Dette er spesielt viktig når X-klient og X-tjener befinner seg på forskjellige maskiner i et nettverk.

14.3 Klienter og tjenere i X

Legg merke til at X-klienten og X-tjeneren kan befinne seg enten på samme maskin eller på forskjellige maskiner et eller annet sted i nettverket – ja, på Internett for den saks skyld.

På din Linux-pc kjører du flere X-klienter og én X-tjener på samme maskin. Kommunikasjonen via X-protokoll foregår da internt på denne maskinen. Dersom X-klient og X-tjener befinner seg på separate maskiner, kjører X-tjeneren på den lokale maskinen du sitter ved, det vil si den som mus og skjerm er koblet til, mens X-klienten kjører på en fjern maskin et eller annet sted i nettverket. Kommunikasjonen mellom disse foregår da via nettverket. Programmet kjører altså på en fjern maskin, og visningen og interaksjonen er på den lokale maskinen.

Du synes kanskje dette bryter med den tradisjonelle oppfatningen av tjeneren som et program på en fjern maskin og klienten som et lokalt program. Dette er jo tilfellet for webtjenere og filtjenere, men slik som vi har beskrevet det for X, fungerer det helt motsatt.

Dersom du tenker på klienten som et program som ber om en tjeneste, og tjeneren som et program som utfører denne tjenesten, blir det enklere å forstå. Klient-tjenerbegrepet dreier seg ikke om hvor i nettverket de er plassert, men om oppgaven de er tildelt. Klienten ber om en tjeneste som tjeneren oppfyller uavhengig av hvor de befinner seg. De kan for eksempel, slik som med X, gjerne befinne seg på samme maskin.

X brukt på denne måten gjør at du fra din Linux-maskin kan benytte deg av programmer som kun lar seg kjøre på bestemte Unix-varianter, og som du dermed ikke kan kjøre på din Linux-maskin.

14.4 X og GUI

Ovenfor har vi sett at X tilbyr den grunnleggende funksjonaliteten, det vil si selve grafikkoperasjonene og håndteringen av mus og tastatur. Det er opp til vindus- og skrivebordsbehandler å legge aktuelle GUI-er på toppen av denne funksjonaliteten. Dette kalles også å legge til “look and feel” på brukergrensesnittet.

Et viktig poeng her er at X ikke legger noen føringer på hvordan brukergrensesnittet skal se ut, og hvordan det skal virke. Det er det helt opp til vindus- og skrivebordsbehandleren å bestemme.

Nedenfor skal vi demonstrere litt av denne fleksibiliteten som finnes i X, ved at vi starter opp forskjellige grafiske brukergrensesnitt og observerer hvordan de virker, og hvordan de ser ut.

14.4.1 Stoppe automatisk oppstart av GUI

Noen ganger ønsker vi ikke å bruke GUI, men heller logge inn i et kommandobasert grensesnitt. Da kan vi seinere velge om vi vil starte opp GUI eller ikke.

Å stoppe direkte login til grafisk miljø gjøres ved å velg menyen *System/Administrasjon/Tjenester*. I lista som kommer fram, fjerner du avkryssingen foran *Grafisk innloggingshåndterer (gdm)*. GUI stopper umiddelbart, og du må trykke **Ctrl+Alt+F1** for å komme til et konsoll i tekstbasert modus.

Du kan starte opp GUI-et igjen manuelt ved å logge inn i tekstbasert modus og deretter bruke kommandoen **startx**, og da eventuelt endre tilbake igjen til automatisk oppstart av GUI. Du kan også starte opp med kommandoen **sudo gdm**.

Utlogging fra det manuelt startede GUI-et skjer nå ved å velge menyen *System/Logg ut*. Merk at du nå kommer tilbake til tekstbasert modus. Du kommer ikke til *Grafisk innloggingshåndterer* igjen, fordi den har du jo stoppet.

14.4.2 Egne tilpassinger for oppstart av GUI

Dersom du ønsker egne tilpasninger for oppstart av GUI, må du opprette filen *.xinitrc* på hjemmemappa og legge inn forskjellige linjer i denne alt etter hva du ønsker å oppnå. Mer om dette i neste kapittel.

Noen operasjoner du kan trenge i kommandogrensesnittet:

- Utlogging med **Ctrl+d** eller med kommandoen **exit**
- Innlogging ved å oppgi brukernavn og passord på vanlig måte
- Slå av maskinen med kommandoen **sudo shutdown -h now**
- Restart av maskinen med **sudo shutdown -r now**
- Manuell oppstart av GUI med **startx**

14.4.3 Oppstart av X uten vindusbehandler

Aller først skal vi starte opp X uten vindusbehandler. Det første du må gjøre, er å opprette en fil, *.xinitrc*, på hjemmemappa di. Da vil X bruke denne filen for å finne ut hvilke programmer som skal startes opp, blant annet hvilken vindusbehandler, eventuelt hvilken skrivebordsbehandler. Du kan også legge inn øvrige programmer du ønsker X skal starte for deg, for eksempel terminalvinduer.

Legg også merke til at noen Linux-distribusjoner krever at filen *.xinitrc* er kjørbart, så dersom du ikke får det til å virke, sjekk at den er kjørbart.

Lag en ny fil *.xinitrc* med innholdet som er vist i den blå rammen nedenfor. Bruk editoren **pico** og start den slik som vist nedenfor. Da opprettes den aktuelle filen automatisk.

```
pico .xinitrc
```

Merk at første tegn i filnavnet er et punktum. Slik må det være fordi dette er en såkalt skjult konfigurasjonsfil.

~/**.xinitrc med kun én X-klient**

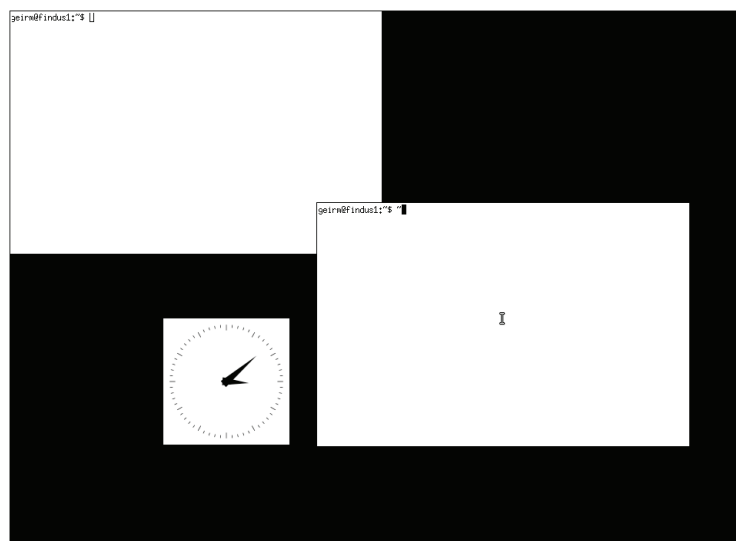
```
# Oppsett av X-miljø og start av X-klienter
xterm &
xclock -geometry +200+400 &
xterm -geometry +400+250
```

Legg merke til at alle kommandolinjene unntatt den siste avsluttes med &-tegnet.

Etter å ha laget *.xinitrc*-filen med innhold og lagret denne starter du opp X med kommandoen

```
startx &
```

X starter nå opp og leser filen *.xinitrc*, og etter en stund ser du en blank skjerm med to terminalvinduer og et klokkevindu. Opsjonen *-geometry* angir plasseringen av vinduet på skjermen i koordinater regnet fra hjørnet øverst til venstre på skjermen.



Figur 14.3 Oppstart av X uten vindusbehandler

Legg merke til at det er ingen rammer rundt vinduene. Det finnes heller ingen metoder for å operere på vinduene, som å flytte vinduer eller å endre størrelse på vinduer. Det er med andre ord ingen vindusbehandler som kjører, og derfor har du ingen muligheter til å operere på vinduene som finnes i brukergrensesnittet, for

eksempel å flytte og å stenge vinduer. Du har kun den basisfunksjonaliteten som X tilbyr.

Siden du ikke har knapper eller menyer for å avslutte X, må du bruke tastekombinasjonen Ctrl+Alt+Backspace for å avslutte X i tilfeller som dette.

En annen mulighet er å gi kommandoen **ps aux** i terminalvinduet og der se etter en linje det står xinit på. Fjern denne xinit-prosessen med kommandoen **kill <pid-nr>**, der pid-nr er prosessnummeret til xinit som du finner i andre kolonne av utlistingen fra **ps**-kommandoen.

14.4.4 Starte opp fullt GUI igjen

Dersom du nå ønsker å starte opp fullt GUI igjen, slik du hadde det til å begynne med må du fjerne filen `.xinitrc` for X. Den enkleste måten å gjøre det på er å endre navnet, for eksempel slik:

```
mv .xinitrc .xinitrc.00
```

Da vil ikke X under oppstarten finne denne filen og vil starte opp helt normalt.

14.4.5 Oppstart av X med enkel vindusbehandler

Rediger på `.xinitrc`-filen slik som vist nedenfor. I tillegg er det nå lagt inn linjer for å starte `xclock` og en ekstra `xterm` slik at vi slipper å starte disse fra kommandolinja, som vi måtte i forrige eksempel.

Siste linje i `.xinitrc`-filen er `twm`, som er en enkel vindusbehandler. Husk `&`-tegn bak alle kommandolinjene unntatt den siste, som starter opp vindusbehandleren.

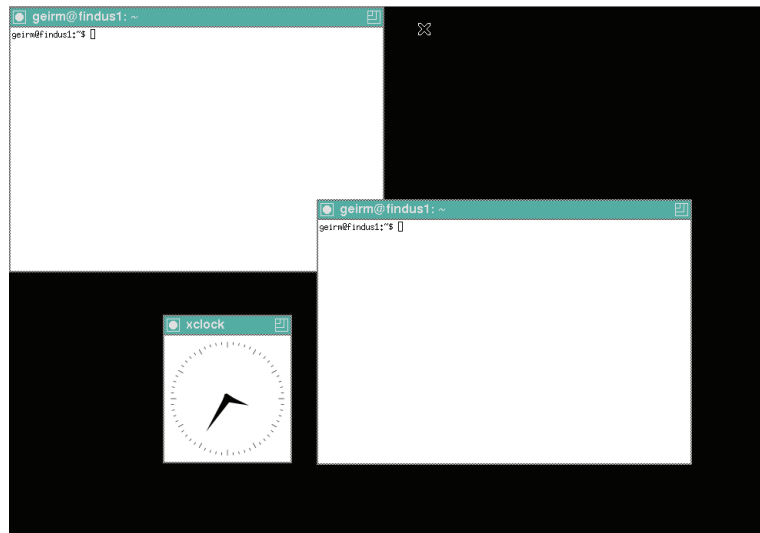
~/`.xinitrc` med flere X-klienter og vindusbehandleren `twm`

```
# Oppsett av X-miljø og start av X-klienter
xterm -geometry +0+0 &
xclock -geometry +200+400 &
xterm -geometry +400+250 &
exec twm
```

Det kan hende at `twm` ikke finnes på Linux-maskinen din. Da må du installere den først. Det enkleste er da å bruke kommandoen

```
sudo apt-get install twm
```

for å installere den. Når det er gjort, prøver du med `startx`-kommandoen på nytt igjen.



Figur 14.4 X med vindusbehandleren twm

Alle programmene som startes opp etter tur, er satt opp med startposisjon. Dersom du ikke angir denne, kommer vinduene til syne som en enkel ramme i tynn strek, og du kan plassere vinduet fritt på skjermen ved å flytte på musa og klikke på museknappen når ønsket posisjon er funnet.

Nå ser du tydelig at vinduene har rammer, og at du enkelt kan flytte vinduene med musa. Ellers fungerer nok vinduene litt forskjellig fra det du er vant til fra Gnome og Windows. Dette er en enkel vindusbehandler, og du kan derfor ikke forvente deg all verden. På venstre museknapp finner du en meny for oppstart av programmer og for utlogging samt for å kunne operere i vinduene.

14.4.6 Interaktiv oppstart av vindusbehandleren

La oss vende tilbake til eksemplet foran og starte opp X uten vindusbehandler.

~/ .xinitrc med flere X-klienter og uten vindusbehandler

```
# Oppsett av X-miljø og start av X-klienter
xterm -geometry +0+0 &
xclock -geometry +200+400 &
xterm -geometry +400+250
```

Fjern siste linje i `.xinitrc`, det vil si den linja som inneholder `exec twm`. Husk å fjerne `&`-tegnet på det som nå blir siste linje i filen. Start opp X med kommandoen **startx &**. I det ene rammeløse terminalvinduet skriver du inn kommandoen **twm &**, slik som vist nedenfor.

```
[kario@findus kario]$ twm &  
[1] 5618
```

Alle vinduene får nå rammer igjen, og du kan operere på vinduene med musa. En vindusbehandler er altså nå kjørende. Siden du startet opp vindusbehandleren som en bakgrunnsprosess, får du også vite prosess-id-en som skrives ut etter kommandolinja. I dette eksemplet er den 5618.

For å avslutte vindusbehandleren bruker du i dette eksemplet **kill**-kommandoen og prosess-id slik som vist nedenfor:

kill <prosess-id>

det vil si at du oppgir prosess-id-en til **kill**-kommandoen. Vindusbehandleren avslutter, og vinduene står tilbake uten rammer. Hvis dette ikke går, må du bruke **kill**-kommandoen på xinit slik som vist foran og dermed ta livet av hele X-sesjonen.

Alle vindusbehandlere kan startes opp på tilsvarende måte. Prøv for eksempel med kommandoen

- **metacity** for å starte opp vindusbehandleren til Gnome (som finnes i Gnome)
- **wmaker** for å starte opp WindowMaker
- **blackbox** for å starte opp BlackBox
- **xfwm4** for å starte opp vindusbehandleren til XFce (som finnes i Xubuntu)
- **icewm** for å starte opp IceWM
- **twm** for å starte opp vindusbehandleren Twm
- **sawfish** for å starte opp Sawfish
- **kwin** for å starte opp KDEs vindusbehandler (som finnes i Kubuntu)
- **fvwm** for å starte den enkle og lite ressurskrevende vindusbehandleren fvwm
- **afterstep** for å starte opp Afterstep vindusbehandler

Den første av disse finnes allerede på maskinen dersom du kjører Gnome, så den kan du prøve med én gang. De øvrige må du installere først med **apt-get**-kommandoen slik:

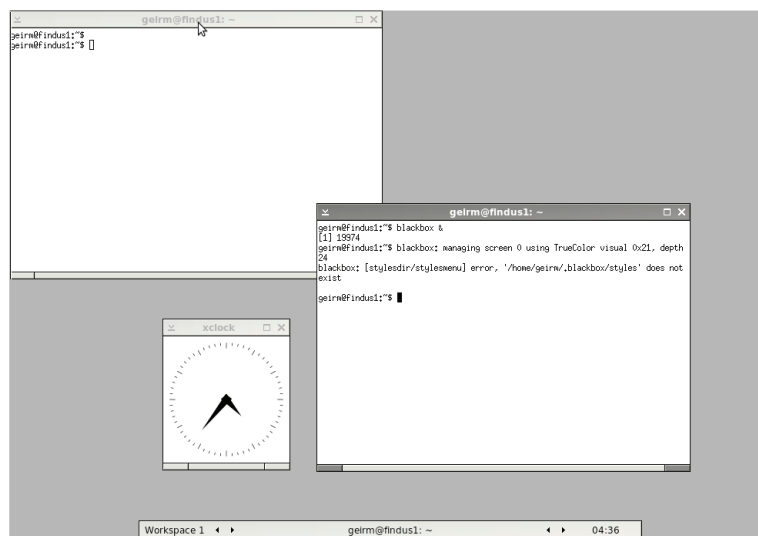
sudo apt-get install <vindusbehandler>

der `<vindusbehandler>` er et av navnene i lista gitt ovenfor.

For noen av vindusbehandlere startes det også ekstra programmer som du ikke får vite prosess-id-en til under oppstart. Da kan det noen ganger være litt vanskelig å bli kvitt hele vindusbehandleren med **kill**-kommandoen. Prøv **ps**-kommandoen for å finne id-en til prosessene. Dersom dette ikke går, kan du trykke tastekombinasjo-

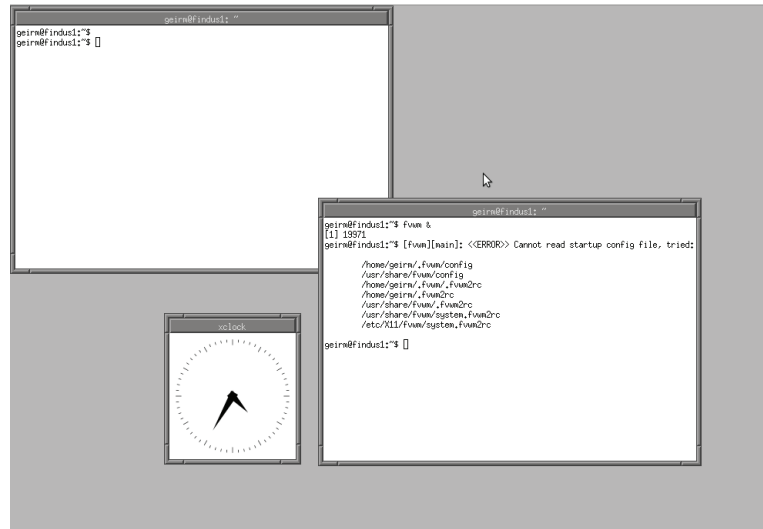
nen `Ctrl+Alt+Backspace` for å avslutte X, eller du kan bruke **kill**-kommandoen på `xinit`-prosessen.

Nedenfor ser du eksempler på vindusbehandlerne `Blackbox` og `fvwm`. `Blackbox` skiller seg fra de andre enkle vindubehandlerne ved at den har en stripe med arbeidsområder nederst.



Figur 14.5 Vindusbehandleren `Blackbox`

Vindusbehandleren `fvwm` er enkel og rett fram uten all mulig stæsj. Videre er den også svært lite ressurskrevende og foretrekkes derfor ofte på eldre maskiner.



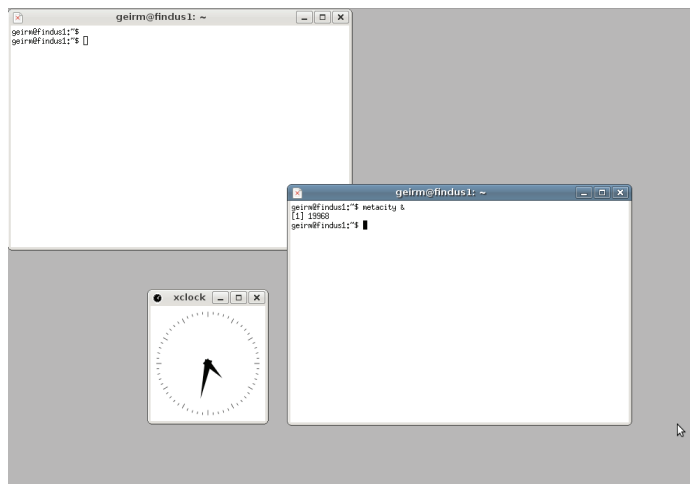
Figur 14.6 Vindusbehandleren fvwm

14.4.7 Skrivebordsbehandleren Gnome og X

Mens vindusbehandleren er et enkelt program for håndtering av vinduer på skjermen, er skrivebordsbehandleren et komplett GUI for Linux. I skrivebordsbehandleren inngår en skrivebordsflate med ikoner, panellinje nederst på skjermen, enkel vindusbehandler, filbehandler m.m. Via disse komponentene får brukeren et bilde av Linux-maskinen, der mapper og filer representeres av ikoner som brukeren kan manipulere via et grafisk brukergrensesnitt.

I standardoppsettet til Gnome startes alle disse komponentene etter tur når du logger inn, slik at du får et komplett skrivebordsoppsett. Men det er ingenting i veien for at du kan starte (og stoppe) disse komponentene hver for seg. Dette viser også at GUI-er for Linux basert på X er svært modularisert bygd opp, altså som separate komponenter.

La oss starte med X uten vindusbehandler, slik som vist i figur 14.3. Start deretter standard vindusbehandler for Gnome med kommandoen **metacity &**. Etter oppstartingen ser skjermen ut som vist i figuren nedenfor.

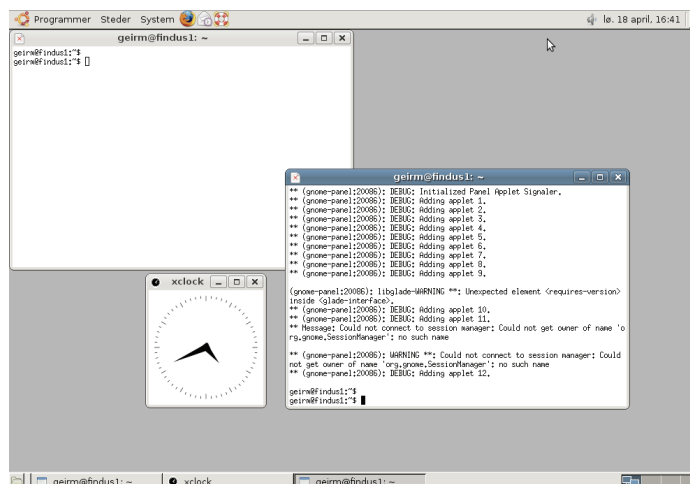


Figur 14.7 X med vindusbehandleren Metacity

Vindusrammene slik du er vant til dem fra Gnome, kan umiddelbart kjennes igjen. Sjekk for eksempel vindusmenyen øverst til venstre.

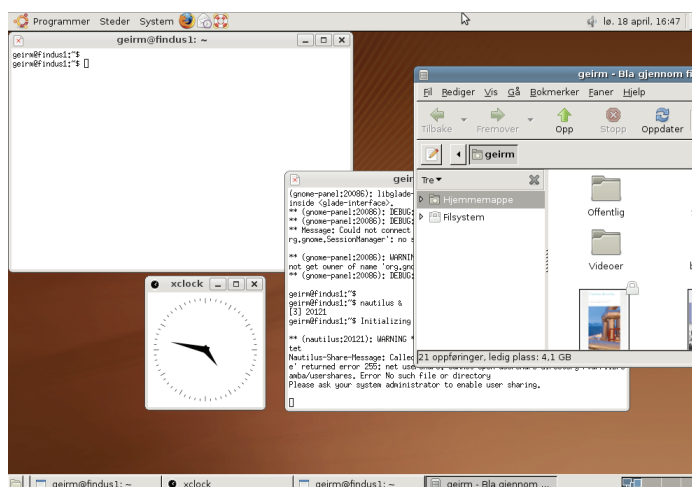
Merk at du ikke ser de øvrige komponentene du er vant til fra Gnome. Disse er ikke startet opp ennå. Kun vindusbehandleren er startet.

Neste trinn er å skrive inn kommandoen **gnome-panel &** i et av terminalvinduene. Panellinjene øverst og nederst på skjermen kommer til syne. Disse er ferdig satt opp med menyer og ikoner slik du kjenner dem fra standardoppsettet til Gnome.



Figur 14.8 X med vindusbehandler og panellinje fra Gnome startet opp hver for seg

I tillegg hører også filbehandleren Nautilus med til skrivebordsmiljøet. Den startes med kommandoen **nautilus &** i et av terminalvinduene. Vinduet til filbehandleren kommer fram på skjermen. Men i tillegg kan du også se at bakgrunnsvinduet på skrivebordet settes. Dette skyldes nok at Nautilus også har ansvar for ikoner som ligger på skrivebordet og dermed er en slags forlengelse av filbehandleren. Du kan for eksempel legge snarveier, mapper etc. på skrivebordet, og det er ting som tilhører filbehandleren.



Figur 14.9 Etter at Nautilus er startet opp

14.4.8 Å starte Gnome fra filen `~/.xinitrc`

Normalt er oppstartingen av GUI-er som for eksempel Gnome ferdig satt opp på maskinen din etter at du har installert Linux. Under installeringen har du for eksempel valgt at maskinen skal bruke grafisk innlogging og automatisk oppstartingen av GUI. Dette kan du seinere endre ved å stoppe prosessen for grafisk innlogging slik som vist i avsnitt 14.4.1 på side 278. Da får du i stedet en tekstbasert innlogging, og oppstart av GUI skjer manuelt med kommandoen **startx &**.

Du kan kombinere den forhåndsdefinerte oppstartingen av Gnome med oppstartingen av egne X-klienter ved å bruke oppstartsfilen `~/.xinitrc` slik som vist nedenfor.

`~/.xinitrc` med flere X-klienter og skrivebordsbehandleren Gnome

```
# Oppsett av X-miljø og start av X-klienter
xterm -geometry +0+0 &
xterm -geometry +400+250 &
xclock -geometry +200+400 &
exec gnome-session
```

Du kan nå starte opp GUI ved å gi kommandoen **startx &**, som leser filen `~/.xinitrc` og starter opp alle programmene som er angitt her. I eksempelfilen ovenfor startes to xterm-vinduer og et xclock-program. Først er vinduene uten rammer, men når Gnome til slutt startes opp med kommandoen **exec gnome-session** slik som vist nederst i filen `.xinitrc`, får vinduene rammer, og i tillegg starter alle de andre Gnome-komponentene.

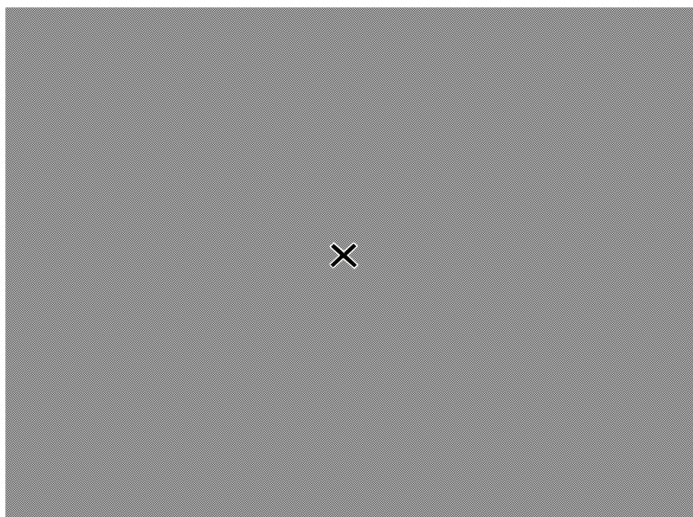
Dersom du ikke ønsker oppstarting av egne programmer slik som vist ovenfor, må du fjerne filen `~/.xinitrc` fra hjemmemappa di. Det kan du for eksempel gjøre ved å skifte navn på filen til `~/.xinitrc00`. Da får du en standard oppstart av Gnome slik Ubuntu har definert oppstarten, uten programmene nevnt i `~/.xinitrc`.

14.4.9 Å starte opp X.org aleine

Du kan gjerne starte opp X aleine, uten noen medfølgende programmer. Dette kan du gjøre med kommandoen **X**. Heng på en `&` dersom du vil ha kommandoprompten fri etter at kommandoen er gitt:

```
X &
```

Skjermen skifter nå fra tekstbasert til grafisk modus. Det eneste du ser, er en farget skjermflate pluss musemerket formet som en x. Ingenting annet er startet opp.



Figur 14.10 Oppstart av X aleine, hvor kun et musemerke formet som en X er synlig

En undersøkelse med **ps**-kommandoen viser at prosessen X kjører. Det er X-tjeneren som nå kjører – X-tjeneren som har ansvaret for skjermen og grafikken og for inndataenhetene mus og tastatur.

Kommandoen **ps** får du kjørt ved å svitsje til virtuell skjerm og kommandomodus med tastekombinasjonen Ctrl+Alt+Fn, der Fn er funksjonstast fra F1 til F6. Tilbake til grafisk modus igjen med tastekombinasjonen Ctrl+Alt+F7.

Du kan avslutte X-tjeneren med **kill**-kommandoen eller bruke tastekombinasjonen Ctrl+Alt+Backspace.

14.4.10 Etterord om X og GUI

Eksempelene vist ovenfor er kun tatt fram for å demonstrere hvordan X og de forskjellige GUI-komponentene virker sammen på skjermen. I ditt normale arbeid på Linux-pc-en bør du bruke de vanlige innloggingsmetodene som Linux-en din er satt opp med. Da kjøres nemlig skallprogrammer som i detalj setter opp og konfigurerer GUI-et du bruker. Det er nødvendig for effektiv og problemfri jobbing i et grafisk brukergrensesnitt.

14.5 Grafisk login-vindu

I eksemplene ovenfor har vi brukt tekstbasert innlogging med etterfølgende oppstartning av X med kommandoen **startx &**. Det finnes grafiske varianter av det vanlige tekstbaserte login-programmet. Dette er programmer som gjør akkurat det samme som login-programmet, det vil si leser brukernavn og sjekker at det finnes på passordfilen, og på tilsvarende måte sjekker at passordet stemmer. Forskjellen er at den grafiske varianten er mer elegant, blant annet med liste over alle brukernavnene, klokke som viser tiden, pluss en del andre muligheter som brukeren selv kan tilpasse.

Disse grafiske login-programmene kalles “Display Managers”, og det finnes som regel flere av dem på Linux-maskinen din. Den opprinnelige som følger med X, er en meget enkel variant som viser to felter midt på skjermen, ett for brukernavn og ett for passord. Dette login-programmet starter du med kommandoen **xdm &**.

De to andre login-programmene er gdm og kdm, for henholdsvis Gnome og KDE. Bokstavene i navnet gdm står for “gnome display manager” og i kdm for “kde display manager”. Du kan gjerne starte opp disse for å se hvordan de ser ut.

Du kan fritt velge hvilket login-program du vil benytte uavhengig av hvilken skrivebordsbehandler du foretrekker.

Gnomes program starter du med kommandoen

```
sudo gdm
```

(sudo fordi programmet må startes som root). Etter noen sekunders oppstartningstid kommer login-vinduet til syne på skjermen. Dette kjenner du igjen fra tidligere innlogginger på Gnome. Du kan gjerne fortsette videre og logge inn på vanlig måte.

Prøv å avslutte login-programmet med tastekombinasjonen Ctrl+Alt+Backspace, slik som du har gjort tidligere for å avslutte X. I dette tilfellet virker det ikke, fordi gdm starter opp igjen automatisk. Det er også virkemåten til det tekstbaserte login-programmet. Etter at du er ferdig, det vil si er logget ut, skal dette starte opp igjen automatisk. Slik er det også med de grafiske login-programmene når systemet er satt opp slik at disse skal brukes i stedet for tekstbasert innlogging.

For å avslutte gdm må du først bruke **ps**-kommandoen for å finne prosess-id-en til gdm. Det gjør du ved å skifte til en virtuell skjerm med tastekombinasjonen Ctrl+Alt+Fn, der Fn er en funksjonstast F1–F6. Prøv flere ganger dersom det ikke virker første gang. Skriv inn kommandoen **ps aux** for å identifisere prosessene gdm. Ta livet av gdm-prosessen(e) med **kill**-kommandoen, og du er tilbake i tekstbasert modus. Husk å skrive sudo foran ps, da det trengs root-privilegier for å ta livet av disse prosessene. Alternativt kan du prøve kommandoen **sudo killall gdm**.

Prøv gjerne også xdm. Det kan hende du må laste ned og installere den først.

14.6 X sett fra innsiden

X har to roller på Linux-maskinen. For det første tilbyr den grunnleggende funksjoner til programmer, vindusbehandlere etc. som har bruk for grafikkressurser av typen tegnefunksjoner, farger, skrifttyper (fonter) etc. Dette har vi allerede sett på. Nå skal vi se på den andre rollen som X har, nemlig som utstyrsdriver, det vil si at den har kontroll over og styring med skjermen, grafikkortet, musa og tastaturet.

Det finnes flere implementasjoner av X. Ubuntu Linux bruker Xorg. Tidligere var XFree86 mye brukt.

14.6.1 Konfigurasjonsfilen til Xorg

Konfigurasjonsfilen til Xorg finner du i mappa */etc/X11/* med navnet *xorg.conf*. Denne filen definerer maskinvaren som brukes av X, det vil si skjerm, grafikkort, mus og tastatur. Denne filen settes opp under installeringen av Linux, og i de fleste tilfellene er det tilstrekkelig, da installasjonsprogrammene etter hvert er blitt svært avanserte og finner det meste av maskinvaren på pc-en din.

Dersom det kun er små justeringer du skal gjøre på filen, kan du redigere på den ved hjelp av en tekstredigerer. Husk å ta en sikkerhetskopi av originalfilen før du begynner å endre på ting. Kall denne filen for *xorg.conf.ORIGINAL* og legg den i samme mappa som de øvrige.

Filen *xorg.conf* består av flere seksjoner, der hver seksjon angir konfigurasjonen for aktuell utstyrsenhet. Hver slik seksjon innledes med navnet "Section" og etterfølges av navnet til den aktuelle seksjonen.

Nedenfor ser du seksjonen for "Inputdevice", først tastatur og deretter mus:

```
Section "InputDevice"
```

```
    Identifier "Keyboard1"
    Driver "Keyboard"
    Option "XkbModel" "pc105"
    Option "XkbLayout" "no"
    Option "XkbOptions" ""
EndSection

Section "InputDevice"
    Identifier "Mouse1"
    Driver "mouse"
    Option "Protocol" "PS/2"
    Option "Device" "/dev/usbmouse"
    Option "Emulate3Buttons"
EndSection
```

I seksjonen for tastatur ser du blant annet

```
Option "XkbLayout" "no"
```

som betyr at maskinen er satt opp med norsk tastatur. Videre ser du under seksjonen for mus at det er snakk om en usb-mus.

Nedenfor finner du seksjoner for skjermen og grafikkortet. Her ser du blant annet at i dette eksemplet er det en flatskjerm med oppløsning på 1400 x 1050 og et grafikkort av typen nVidia.

```
Section "Monitor"
    Identifier "monitor1"
    VendorName "Generic"
    ModelName "Flat Panel 1400x1050"
    HorizSync 31.5-90
    VertRefresh 59-75
EndSection

Section "Device"
    Identifier "device1"
    VendorName "nVidia Corporation"
    BoardName "NVIDIA GeForce2 DDR (generic)"
    Driver "nvidia"
    Option "DPMS"
EndSection
```

Selve skjermen kan kjøres i forskjellige modi. Fargedybden er satt til 24. Det betyr "true colors", og i antall er det snakk om 16 millioner farger. X vil da velge underseksjonen med "Depth 24". Der velges den beste oppløsningen som skjermen greier å vise fram.

De andre skjermoppløsningene gitt nedenfor kan du skifte mellom med tastekombinasjonen Ctrl+Alt+Num-pluss og Ctrl+Alt+Num-minus, det vil si Ctrl+Alt og pluss- eller minustast på den numeriske delen av tastaturet.

```
Section "Screen"
    Identifier "screen1"
    Device "device1"
    Monitor "monitor1"
    DefaultColorDepth 24
```

```

Subsection "Display"
  Depth 24
  Modes "1400x1050" "1280x1024" "1024x768" "800x600" "640x480"
EndSubsection
EndSection

```

Husk at du normalt ikke trenger å lage denne filen selv. Et utkast settes opp under installeringen, og endringer kan du vanligvis ordne ved hjelp av konfigurasjonsverktøyet for den Linux-en du bruker, slik som angitt ovenfor.

14.6.2 Kommandoen `startx`

Kommandoen `startx` har vi brukt mange ganger allerede i eksemplene foran for å starte opp X på maskinen. Denne kommandoen er et skallprogram du finner på mappa `/usr/bin/`. Oppgaven til skallprogrammet er å starte opp X. Først sjekker programmet om det finnes en fil `~/.xinitrc`. Dersom den finnes, brukes de oppstarts-kommandoene som finnes der. Hvis den ikke finnes, går programmet videre og sjekker en del globale oppstartsfiler og bruker informasjonen som finnes der, for oppstarting av X.

Mer informasjon om kommandoen `startx` finner du på manualsiden med kommandoen `man startx`.

14.7 Skrivebordsbehandler versus vindusbehandler

Gnome er et komplett skrivebordsmiljø for Linux, på lik linje med skrivebordsmiljøene du finner for Macintosh og for Microsoft Windows.

Gnome er et åpent system, og det er fritt fram for deg å bruke systemet. Bak Gnome står flere hundre utviklere spreddt over hele verden som samarbeider via nettet, og som i stor grad jobber på ideell basis. Kildekoden er tilgjengelig for alle, og det er fritt fram dersom du ønsker å modifisere koden og dermed lage dine egne varianter av Gnome. Det eneste som kreves, er at du oppfyller kravene for åpen programvare som er nedtegnet i GNU General Public License, kalt GPL (mer info om dette på www.gnu.org). Gnome står for GNU Network Object Model Environment.

Hensikten med et skrivebordsmiljø av typen Gnome er tosidig:

- Å tilby et enkelt og intuitivt grafisk grensesnitt mot maskinens ressurser. Dette betegnes ofte for “look and feel” og refererer til vinduene på skjermen og mus og tastatur på bordet.
- Å tilby et komplett utviklingsmiljø for applikasjoner slik at alle Gnome-applikasjoner får et grafisk brukergrensesnitt basert på samme “look and feel”.

Gnome er ett av flere skrivebordsmiljøer. KDE er et annet med tilsvarende funksjonalitet. Ubuntu Linux kommer i flere versjoner. Standardversjonen bruker Gnome, men du kan også velge Kubuntu, som kjører KDE som standard skrivebordsmiljø.

Om ønskelig kan du også installere det andre skrivebordsmiljøet i tillegg, slik at du ved innlogging kan velge om du skal kjøre Gnome eller KDE (eller et annet skrivebordsmiljø eller vindussystem).

Gnome og KDE er de dominerende skrivebordsmiljøene for Linux. Det betyr at uansett hvilken Linux-distribusjon du velger, så vil du treffe på Gnome eller KDE og kan velge hvilket av disse skrivebordsmiljøene du vil bruke. Hvilken Linux-distribusjon du velger, har dermed ikke så mye å si når det gjelder brukermiljøet. Da står det mellom Gnome og KDE, pluss en del andre enklere vindusbehandlere.

14.7.1 Skrivebordsbehandler

Som bruker av et grafisk brukermiljø av typen Gnome og KDE er det viktig å være klar over hvordan disse er bygd opp. Dermed kan du lettere forstå hva som skjer når komponenter skiftes ut, og når du veksler mellom forskjellige systemer. Grafiske brukermiljøer i Linux kan nemlig tilpasses i det uendelige, og komponenter kan skiftes ut etter eget ønske.

Det grafiske brukermiljøet, det vil si skrivebordsbehandleren i Gnome, består av følgende komponenter:

- *Skrivebordsflaten* (eng.: desktop area): Denne komponenten håndterer ikoner på skrivebordet, funksjoner som kan utføres på skrivebordet, bakgrunnsbilder, program- og filikoner som legges på skrivebordet etc.
På skrivebordsflaten finner du ikoner, mapper og filer. Skrivebordsflaten administreres derfor av filbehandleren og startes derfor opp samtidig med den.
- *Vindusbehandler* (eng.: window manager; programnavn: metacity): Denne komponenten håndterer vinduers utseende og virkemåte. Med utseende menes utformingen til vindusrammer, farger, knapper på tittellinje etc. Med virkemåte menes hvilke operasjoner som er tilgjengelige på vinduer, og hvordan disse fungerer, det vil si hvordan åpne, lukke, flytte, endre størrelse etc. på vinduer. Videre dreier det seg om bruken av rullefelt på vinduer.
Gnome kan gjerne brukes sammen med andre vindusbehandlere. Mer om dette seinere i kapitlet.
- *Panellinjene* (programnavn: gnome-panel): Panellinjene finner du øverst og nederst på skjermen og består av menyer, programikoner, startede programmer etc.
- *Filbehandleren Nautilus* (programnavn: nautilus): Dette er et program for å navigere i filsystemet til Linux, vise innholdet til kataloger og til filer, foreta operasjoner som å opprette, kopiere, slette og å gi nytt navn til filer og kataloger.

Komponentene som er nevnt ovenfor, er komponenter som typisk inngår i skrivebordsbehandlere. Dersom du ikke finner disse komponentene på skrivebordet ditt, kjører du etter all sannsynlighet bare en enkel *vindusbehandler*.

Ideen bak en skrivebordsbehandler er å tilby et grensesnitt der programmene har samme virkemåte med hensyn til oppstart, oppsett av menyer, knapper, vinduer etc. Via skrivebordsbehandlere av typen Gnome og KDE opplever brukeren et helhetlig brukergrensesnitt der “look and feel” er konsistent gjennom hele systemet.

Derfor består skrivebordsbehandleren også av et programutviklingsmiljø, det vil si et komplett sett av biblioteker som programutviklere kan benytte for å lage programmer som oppfører seg slik et Gnome-program skal gjøre, og som dermed gir en konsistent “look and feel” for alle Gnome-programmer.

14.7.2 Vindusbehandler

Vindusbehandleren er kanskje det programmet du jobber mest med i skrivebordsmiljøet. Både utseendet og interaktiviteten til skrivebordet er bestemt av vindusbehandleren. Med *utseende* tenker en på slike ting som utforming av vindusrammer og utseende til knapper. I skrivebordssammenheng kalles dette forskjellige tema, og det er helt vanlig å kunne skifte tema for skrivebordet. Med *interaktivitet* menes hvordan knappene virker, hvordan menyer aktiviseres, hvordan vinduer åpnes, flyttes, lukkes etc.

I denne sammenhengen er det viktig å skille vindusbehandleren fra selve applikasjonen:

- Applikasjonen kjører inne i vinduet med sitt eget utseende og sin egen interaktivitet uavhengig av vindusbehandleren. Applikasjonen kan derfor bare påvirke innholdet i vinduet, og ikke selve vinduet.
- Vindusbehandleren, derimot, har ansvaret for utseendet og virkemåten til vinduene.

Det betyr at vi gjerne kan endre virkemåten til vindusbehandleren, eller skifte til helt ny vindusbehandler, uten at dette påvirker innholdet i vinduene, det vil si applikasjonen.

I forrige kapittel så vi at Gnome og KDE var typiske skrivebordsbehandlere. Når det gjelder vindusbehandlere, så finnes det en lange rekke av disse, blant annet WindowMaker, Enlightenment, BlackBox, XFce, IceWM, AfterStep, twm, fvwm og ikke å forglemme metacity, som er standard vindusbehandler til Gnome. KDE bruker en vindusbehandler kalt kwin.

Det kan ofte være vanskelig å skille mellom skrivebordsbehandler og vindusbehandler. La oss slå fast følgende:

- Gnome og KDE er skrivebordsbehandlere og er derfor et komplett skrivebordsmiljø med flere komponenter, blant annet med standard vindusbehandler, som er metacity for Gnome og kwin for KDE.
- Gnome og KDE kan gjerne skifte ut standard vindusbehandler med en annen, for eksempel fvwm eller WindowMaker.

- Det er ikke nødvendig å kjøre et komplett skrivebordsmiljø, det vil si du trenger ikke å kjøre skrivebordsbehandlere av typen Gnome og KDE. Du kan gjerne kjøre vindusbehandleren `aleine`.

Skrivebordsmiljøet i Linux er altså ganske fleksibelt. Du kan selv velge hvilken skrivebordsbehandler og hvilken vindusbehandler som skal brukes, for eksempel kjøre Gnome sammen med vindusbehandleren `fvwm`. Ja, du kan endog splitte det opp videre og for eksempel velge hvilke panellinjer som skal brukes, nederst på skjermen.

Alle komponentene i det grafiske brukergrensesnittet er separate programmer og kan startes og stoppes etter behov.

Merk også at vi her har beskrevet ytterpunktene, der et komplett skrivebordsmiljø befinner seg i det ene ytterpunktet og en enkel vindusbehandler befinner seg i det andre ytterpunktet. Når du prøver ut mange av vindusbehandlerne som er omtalt i dette kapitlet, vil du finne at mange av dem befinner seg inn mot midten av denne skalaen, da de inneholder et skrivebord med ikoner, de inneholder menyer etc., men mangler ofte et utviklingsmiljø som garanterer konsistent “look and feel”.