



Norges teknisk–naturvitenskapelige
universitet
Institutt for datateknologi og
informatikk

TDT4102 Prosedyre- og objektorientert programmering Vår 2019

Øving 0 for Linux

Frist: Ingen (frivillig øving)

Mål for denne øvingen:

- Bli kjent med programmeringsverktøy
- Lage et første program med Ubuntu
- Kunne laste ned og kjøre eksempelprogram fra forelesningene med Ubuntu
- Lage et første program kun med teksteditor og kompilator

Denne øvingen er en veiledning i å installere en programmeringsomgivelse slik at du kan skrive, redigere, compilere, debugge og kjøre et C++ program. **Det er nødvendig å gjennomføre og mestre det meste av det som gjennomgås i denne øvingen for å kunne utføre de obligatoriske øvingene.**

Hvis du kjører GNU/Linux eller en annen form for Unix-liknende operativsystem (macOS har egen øving 0), er denne øvingen en generell veiledning til hvordan fagets øvinger *kan* kompiles og kjøres. For å gjøre det enkelt er det i denne øvingen også kun installering i Ubuntu som vil vises, det antas at de som velger å benytte andre varianter av linux kjenner til hvordan et bibliotek lastes ned, kompiles og installeres. **Merk at de to primære platformene som anbefales og støttes i faget TDT4102 er Windows PC og Mac.** De aller aller fleste studentene i faget benytter PC eller Mac, og *fagstaben har ikke kapasitet til å gi veiledning i bruk av Linux.* Hvert år har vi et mindre antall studenter i faget som er godt kjent med Linux, og dette notatet er ekstra hjelp til disse, selv om disse studentene pleier å klare seg selv. Notatet er *ikke* en oppfordring til PC eller Mac brukere til å skifte over til Linux.

Programvare som installeres i denne øvingen er bl.a. `clang`, `make` og `fltk`.

Aktuelle kapitler i boka:

- Kapittel 0, 1 og 2 i Programming – Principles and Practice Using C++ (Second Edition)

Installasjon av byggeverktøy og fltk

```
sudo apt install build-essential clang clang-6.0 libfltk1.3-dev
```

Linjen over installerer bl.a. make, clang++ og fltk 1.3.

Boken og øvingene bygger på en headerfil og en wrapper for et grafikkbibliotek. Headerfilen `std_lib_facilities.h` vil skjule noe av kompleksiteten og gjøre programmering i C++ tryggere for den første halvdel av kurset. Wrapperen `graph_lib` gjør det mulig å bytte ut det underliggende grafikkbiblioteket (fltk) uten at applikasjoner som benytter `graph_lib` må kompileres på nytt eller endres.

Last ned `TDT4102_Linux.zip` fra blackboard. Pakk ut innholdet og gå til roten av den nye stien:

```
unzip TDT4102_Linux.zip
cd TDT4102_Linux
```

Kompilering av `graph_lib`

Du bestemmer selv hvordan du ønsker å plassere headerfiler og bibliotek. Et alternativ er å kompilere `graph_lib` og legge det i et bibliotek. Det følger med en makefil som gjør dette for deg.

```
cd Graph_lib
make Graph_lib.a # makeregel som bygger og pakker graph_lib
cd ..
```

Nå skal du ha en filstruktur der du kan se filstiene `Graph_lib`, `include` og `example`. `example` er et eksempel på hvordan en øving kan se ut. For å verifisere at alt fungerer som det skal, kjør:

```
cd example
make run
```

Du skal nå se en rød trekant. Hvis dette fungerte uten problemer kan du rydde opp etter kompileringen ved å skrive:

```
make clean
```

Deretter kan du gå opp et nivå og kopiere `example`-mappen som utgangspunkt for andre øvinger. Øving 3 kan f.eks. opprettes som følger:

```
cd .. # Går opp et nivå
cp -r example Oving3 # Kopier hele mappen example og gir kopien navnet Oving3
```

Som standard vil navnet til programmet som kompileres vha. makefilen i `example`-prosjektet være `example`. Dette kan endres i makefilen der det er henvist med en kommentar i første linje.