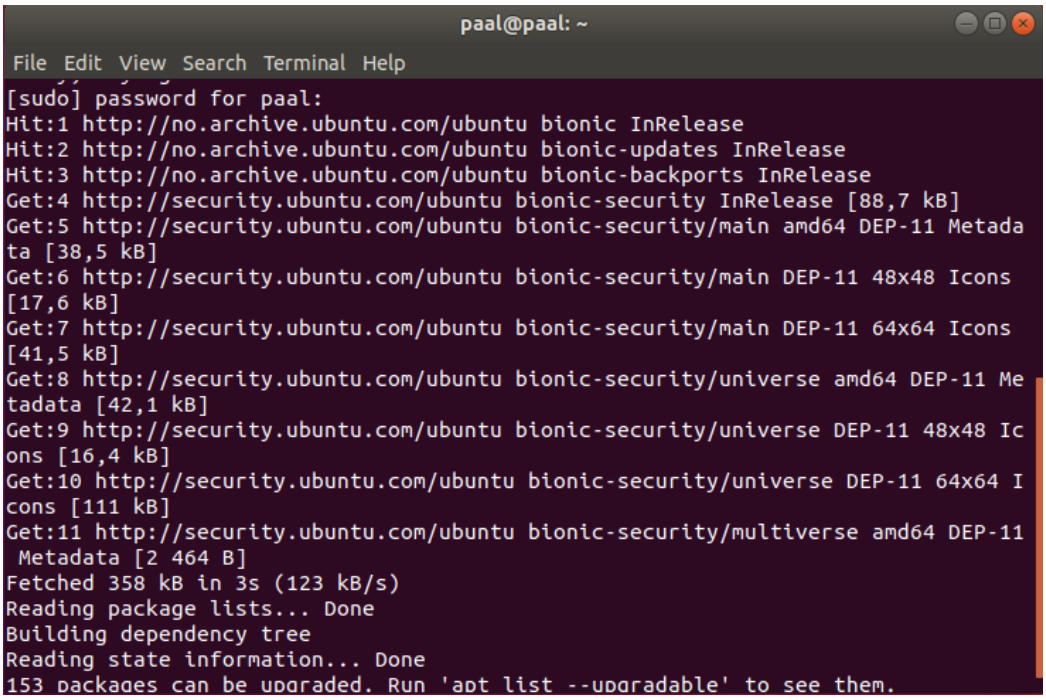
Lab 3 - linux

Oppsett:

Eg installerte Ubuntu på ein gammal pc eg hadde med og bruka Ubuntus sin egent installer.

https://ubuntu.com/download/desktop



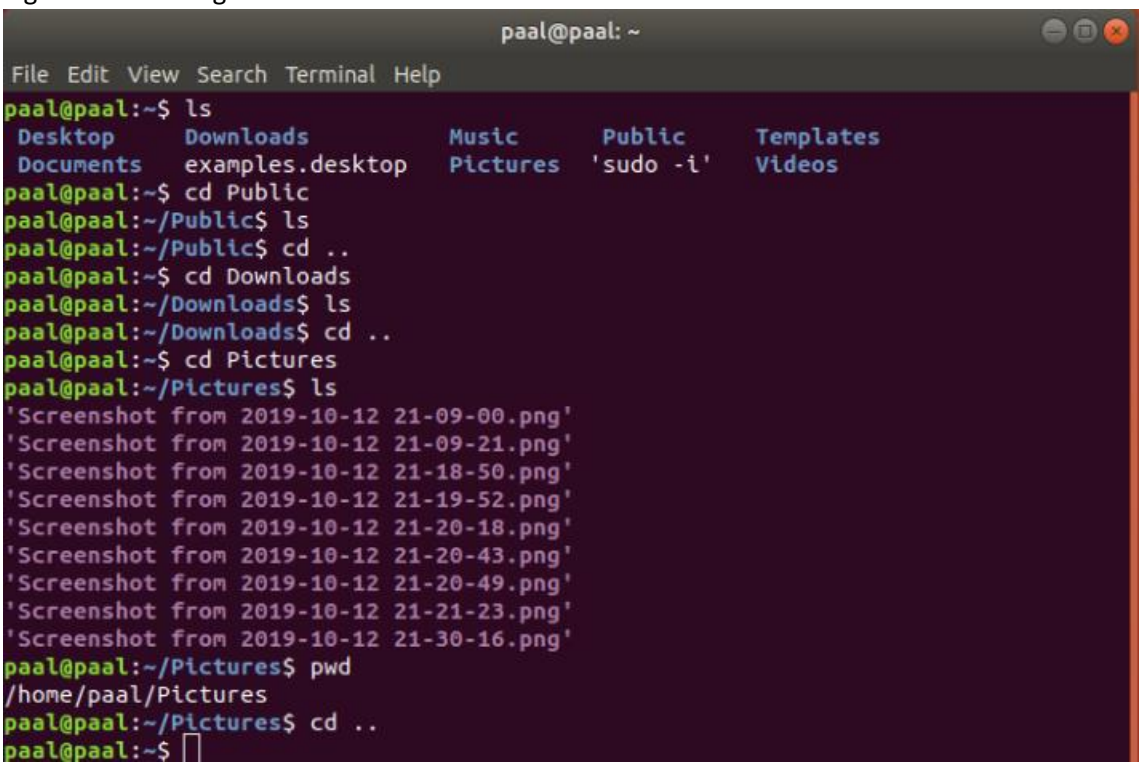
Brukte commanden «sudo apt update» for så og kjøra «sudo apt upgrade» og som du ser så blei 153 packages oppgradert.

Grunnleggande bruk:

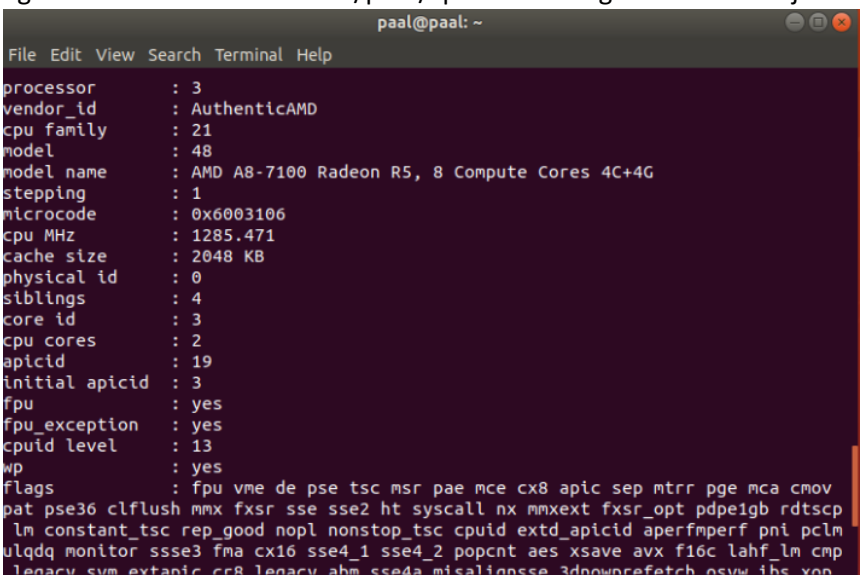
1)

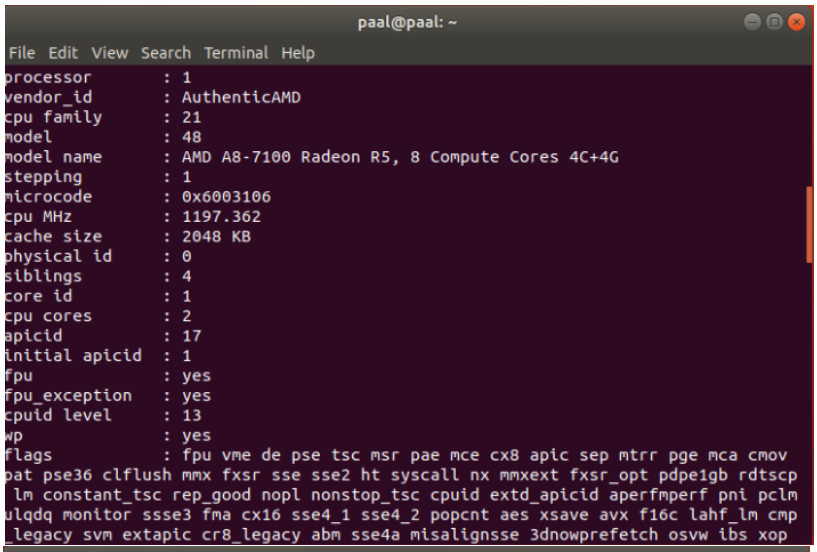
Med å bruka ‘’ls’’ kan eg sjå kva mapper/filer som er inne i mappa eg er i. ‘’pwd’’ bruker eg for å sjå fil veien til der eg er. ‘’cd «fil name»’’ bruker ein for og komme seg inn i fila ein vil og ‘’cd ..’’ bruker ein for

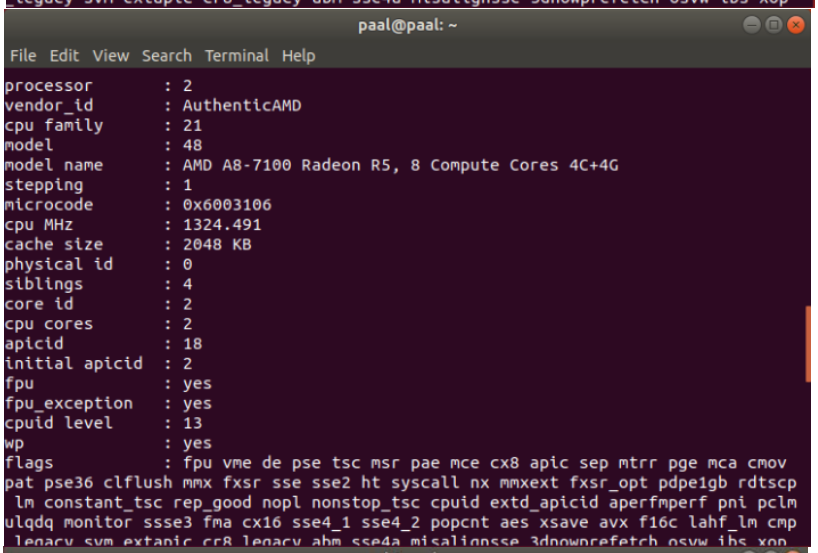
å gå tilbake et steg.

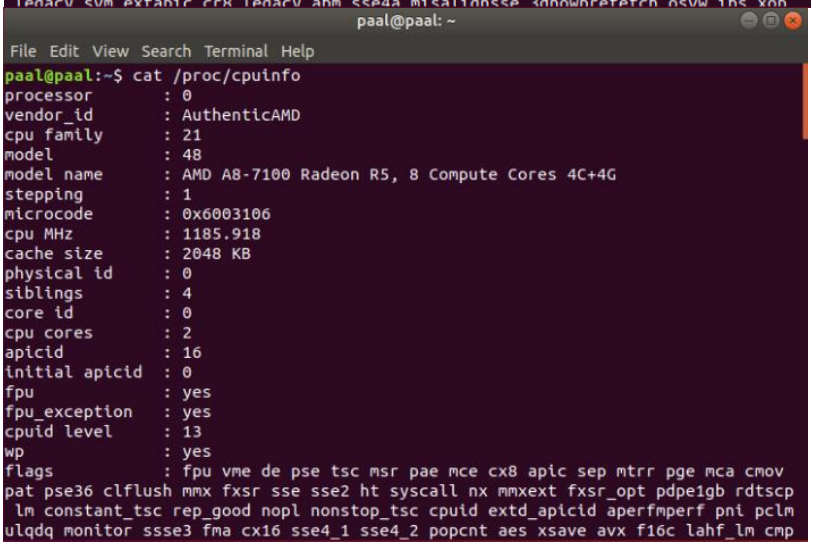


2) Eg brukte kommandoen «cat /proc/cpuinfo» for og finna informasjon om CPUen min

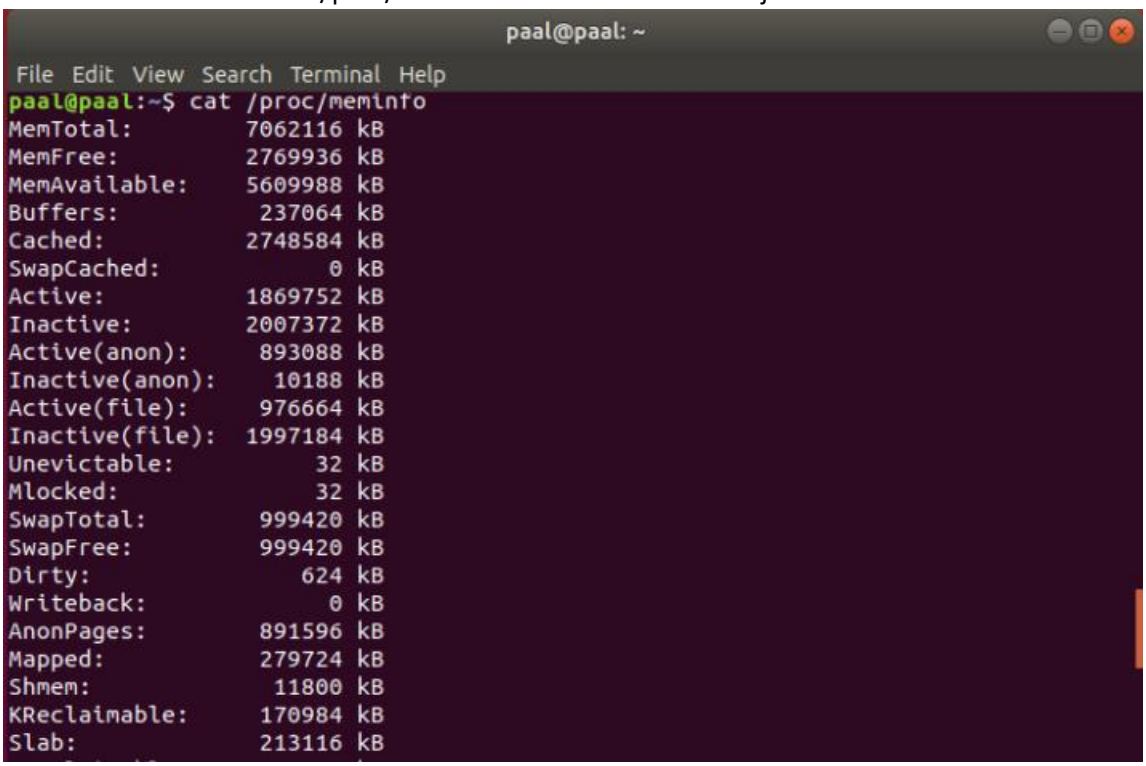



paal@paal: ~
File Edit View Search Terminal Help
processor : 1
vendor_id : AuthenticAMD
cpu family : 21
model : 48
model name : AMD A8-7100 Radeon R5, 8 Compute Cores 4C+4G
stepping : 1
microcode : 0x6003106
cpu MHz : 1197.362
cache size : 2048 KB
physical id : 0
siblings : 4
core id : 1
cpu cores : 2
apicid : 17
initial apicid : 1
fpu : yes
fpu_exception : yes
cpuid level : 13
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclm
ulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 popcnt aes xsave avx f16c lahf_lm cmp
legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs xop


paal@paal: ~
File Edit View Search Terminal Help
processor : 2
vendor_id : AuthenticAMD
cpu family : 21
model : 48
model name : AMD A8-7100 Radeon R5, 8 Compute Cores 4C+4G
stepping : 1
microcode : 0x6003106
cpu MHz : 1324.491
cache size : 2048 KB
physical id : 0
siblings : 4
core id : 2
cpu cores : 2
apicid : 18
initial apicid : 2
fpu : yes
fpu_exception : yes
cpuid level : 13
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclm
ulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 popcnt aes xsave avx f16c lahf_lm cmp


paal@paal: ~
File Edit View Search Terminal Help
paal@paal:~$ cat /proc/cpuinfo
processor : 0
vendor_id : AuthenticAMD
cpu family : 21
model : 48
model name : AMD A8-7100 Radeon R5, 8 Compute Cores 4C+4G
stepping : 1
microcode : 0x6003106
cpu MHz : 1185.918
cache size : 2048 KB
physical id : 0
siblings : 4
core id : 0
cpu cores : 2
apicid : 16
initial apicid : 0
fpu : yes
fpu_exception : yes
cpuid level : 13
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclm
ulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 popcnt aes xsave avx f16c lahf_lm cmp

3) Bruker kommandoen «cat /proc/meminfo» for å finna informasjon om minne

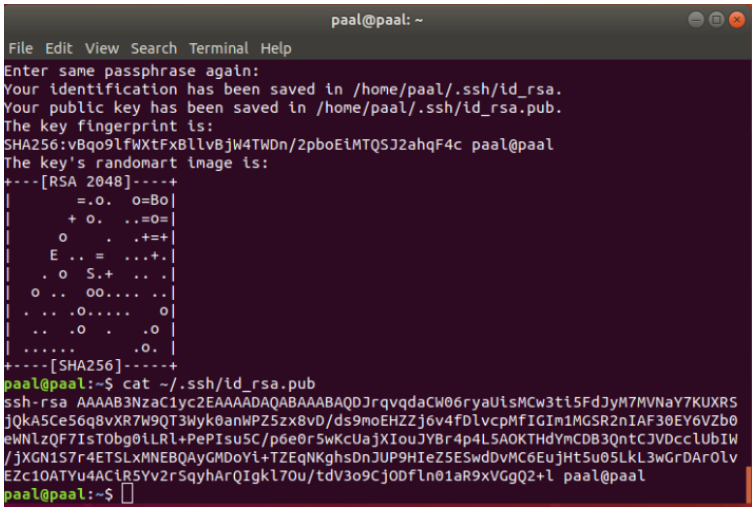


Installer programmer

1) Brukte kommandoe ‘’sudo apt install «program namn»’’ for og installera dei forskjellige programma ein treng seinare i oppgava.
2) Her gjorde eg som første gang eg konfigurerte git

```
git config --global user.name "<your full name>"
git config --global user.email "<your UiA e-mail address>"
```

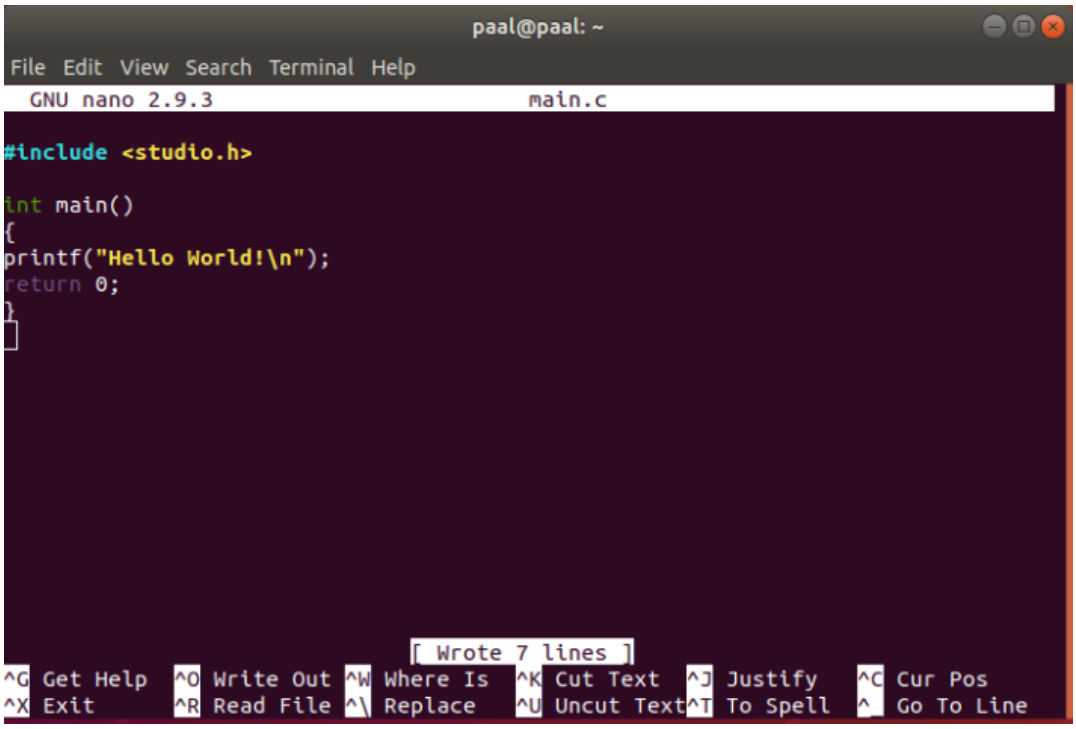
og for å sjekka at alt stemte bruke eg ‘’cat ~/.gitconfig’’. så genererte eg ein nøkkel med ‘’ssh-keygen’’ og fan nøkkelen med ‘’ «cat ~/.shh/id_rsa.pub’’



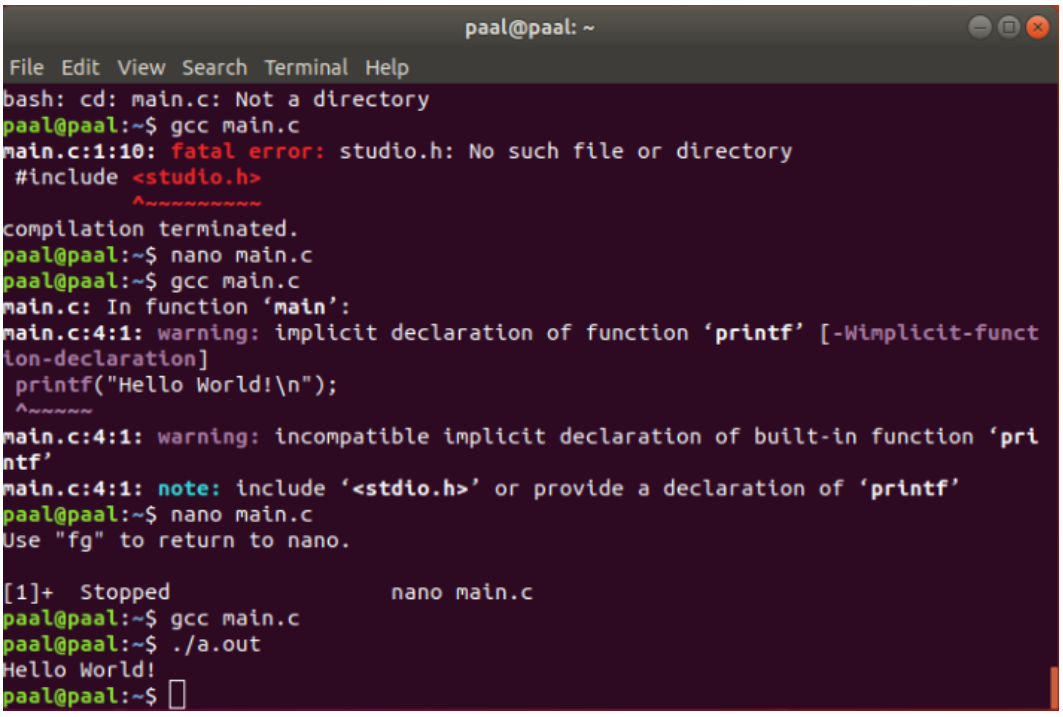
så kopierte eg nøkkelen og la den i «bitbucket»

Bygg et program med kompilator direkte

1) Kjørte ‘’sudo apt install nano’’ men viste seg eg allereie hadde programmet.
2) Brukte ‘’cd’’ så ‘’pwd’’ og fan ut at eg var i ‘’home/paal’’
3) Så brukte eg kommandoe ‘’nano main.c’’



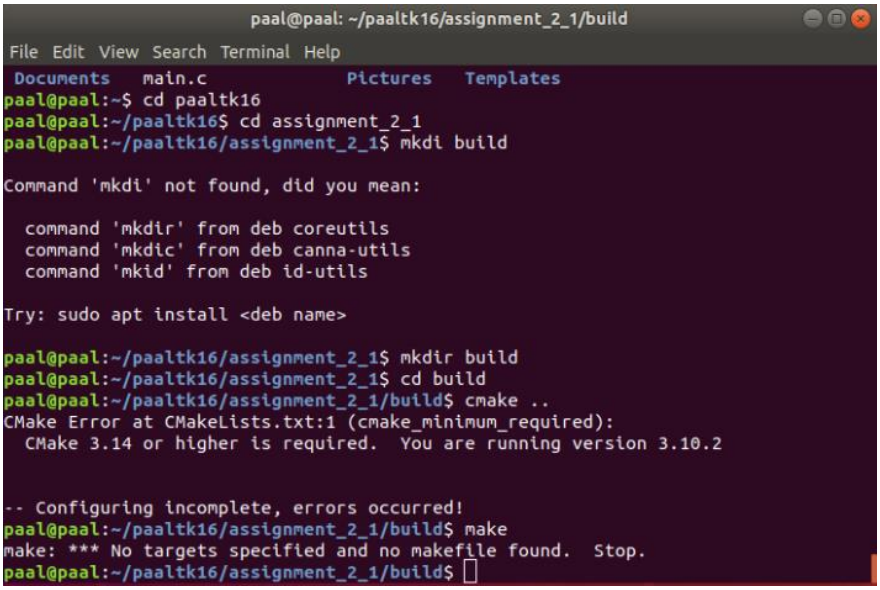
4) Brukte ‘’CTRL + S’’ som du ser ovanfor brukte eg 7 linjer. Så ‘’CTRL + X’’ for å gå utav nano.
5) Kjørte så ‘’gcc main.c’’ så ‘’./a.out’’ som ein ser nedanfor.



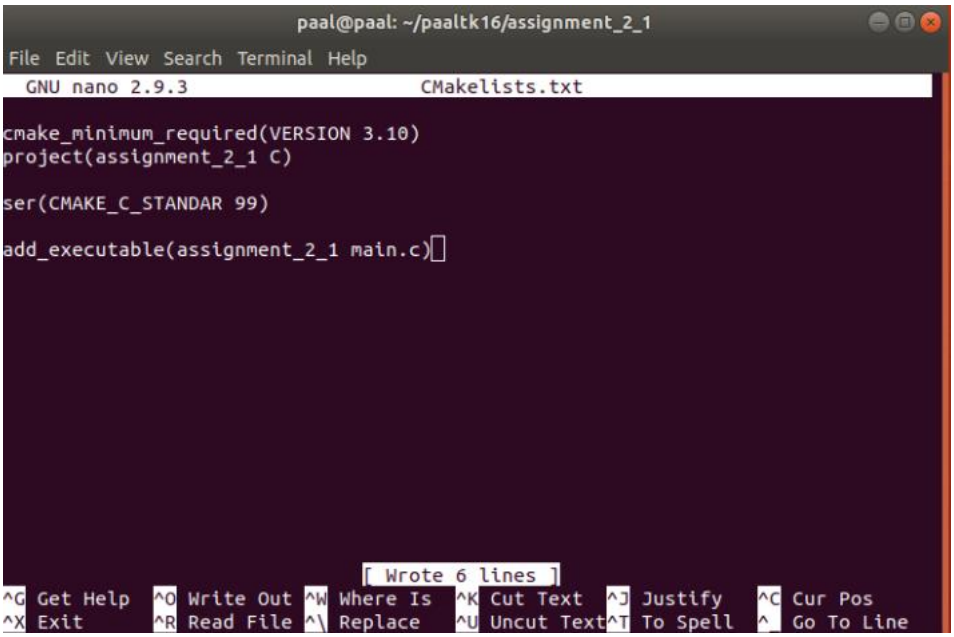
6) GCC (GNU Compiler Collection) er ein rekka med kompilatora som støtte programspråka: C, C++, Objective-C, Fortran, Ada, Go, D. Støtte og biblioteka for språk som «libstdc++, etc». Systeme GNU var bygga og er eit heilt fritt software som skal respektera brukaren sin friheit. [1]

Bygg eit program med CMake og make

1) Eg gjekk innpå «Bitbucket» og fan linken min. så bruke eg kommandoen ‘’git clone «Bitbucket linken»’’
2) Eg gjorde så som ein ser på bilde nedanfor

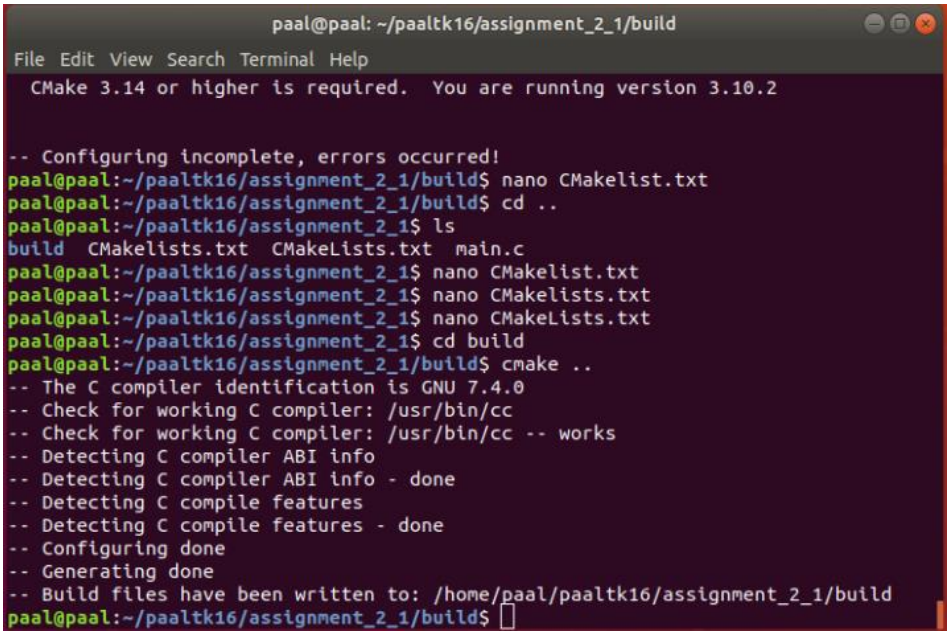


Som ein ser så fekk eg at eg hadde feil versjon så eg brukte kommandoen ‘’nano CMakeList.txt’’ (NB! Husk store og små bokstaver ellers kommer ein ikkje inn på rett plass) sånn at eg kan endra minimums versjonen som krevs for å kjøra.

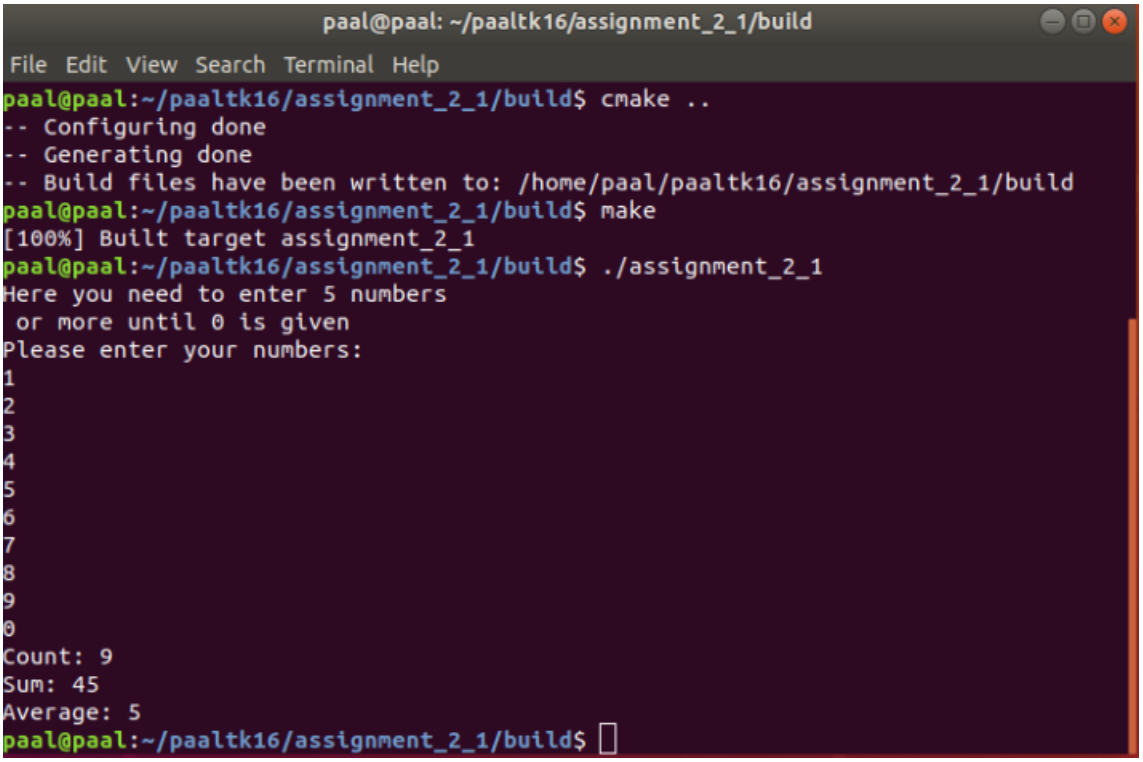


som ein ser her gjorde eg feilen med å gløymma stor ‘’L’’ i CMakeList.txt, men eg gjekk tilbake og endra på da så fort eg fan ut.

Så prøvde eg og kjøra ‘’cmake ..’’ igjen



Og brukte kommandoen ‘’make’’ for å bygga programmet og ‘’./assignment_2_1’’ for å kjøra programmet



Cmake plassere enkle konfigureringsfile som har namnet ‘’CMakeLists.txt’’ inn i alle «source directory». Detta blir brukt til å generera/laga bygg fila «makefiles». Den generere/lagar også bygg innstillinga. [2]

Make lese file kalla ‘’Makefiles’’ frå «source code», disse filene fortell korleis programma skal bli utleid. Detta lar den automatisk bygga program og bibliotek som kan bli eksekvert. [3]

Referansar:

[1] G. Team, «GCC.GNU,» 19 August 2019.

https://gcc.gnu.org/. [Funnet 11 Oktober 2019].

[2] CMake, «CMake,»

https://cmake.org/overview/. [Funnet 11 Oktober 2019].

[3] «Wikipedia,» 21 September 2019.

https://en.wikipedia.org/wiki/Make_(software). [Funnet 11 Oktober 2019].