

# Įvadas į C++ programavimą Linux aplinkoje

## 1. Darbo pradžia

Prisijungus prie savo darbo srities, pasileiskite *shell* programą, kuri priima ir interpretuoja vartuotojo komandas. *Shell* programą galima pasileisti grafinėje darbo aplinkoje taip:

*Utilities -> Shells -> Sh*

Atsiradusiame lange toliau galėsite įvesti įvairias operacinės sistemos komandas ir paleisti programas.

Pastaba: pereikite į *Desktop* katalogą ir susikurkite naują katalogą, kurio pavadinimas sutampa su jūsų pavarde. Toliau savo failus kurkite tik šiame kataloge:

```
$ cd Desktop
$ mkdir jusu_pavarde
$ cd jusu_pavarde
```

Failams redaguoti rekomenduojame naudoti tekstų redaktorių *NEdit*, kuris paryškina C kalbos sintaksę:

*Editors -> Debian -> NEdit*

## 2. C/C++ programų kompiliavimas bei vykdymas

Kompiliuojant C++ programas (failo priesaga .cpp) yra naudojama komanda *g++*:

```
$ g++ failo_vardas
```

Teisingai sukompiliavus gaunamas vykdomasis failas, kurio pavadinimas pagal nutylėjimą yra *a.out*. Programų objektiniai moduliai gaunami nurodžius raktą *-c*:

```
$ g++ -c failo_vardas
```

Objektiniai moduliai turi priesagą *.o*. Norimas vykdomojo failo vardas gali būti nurodytas su raktu *-o*:

```
$ g++ failo_vardas -o vykdomasis_failas
```

Kompiliuojant iš kelių failų sudarytą programinę sistemą, jų vardai išvardinami komandinėje eilutėje:

```
$ g++ -c failas1 failas2 failas3
```

## 1 pavyzdys: pranešimo išvedimas

```
/* failas helloworld.cpp */
#include <iostream>
using namespace std;

int main (int argc, char *argv[]) {
    cout << "hello, world\n";
    return(0);
}
```

Ši programa naudoja išvedimo srautą *cout*, aprašytą antraštės faile *iostream.h*, kad išvestų paprastą pranešimą į standartinį išvedimo įrenginį.

Ši programa gali būti sukompiliuota taip:

```
$ g++ helloworld.cpp -o helloworld
```

Vykdomasis failas *helloworld* įvykdomas taip:

```
$ helloworld
```

## 2 pavyzdys: įvedimas, skaičiavimas ir išvedimas

```
/* failas suma.h */
int suma (int a, int b);
//-----
/* failas suma.cpp */
#include "suma.h"
int suma (int a, int b) {
    int c;
    c = a + b;
    return c;
}
//-----
/* failas calculate.cpp */
#include <iostream>
#include "suma.h"
using namespace std;
int main (int argc, char *argv[]) {
    int a, b, c;
    cout << "Iveskite a ir b:\n";
    cin >> a >> b;
    c = suma(a,b);
    cout << "Suma: " << c << endl;
    return(0);
}
//-----
```

Ši programa paprašo įvesti du skaičius, suskaičiuoja jų sumą ir išveda rezultatą.

Kompiliuojama taip:

`g++ calculate.cpp suma.cpp -o calculate`

Vykdomasis failas *calculate* įvykdomas taip:

`$ calculate`

Keletas naudingų `g++` kompiliatoriaus raktų:

- *I katalogas* Nurodomas katalogas, kuriame kompiliatorius turi ieškoti .h failų.  
- *lvardas* Nurodomas prijungiamos bibliotekos vardas, šiuo atveju libvardas.a

Daugiau informacijos galite rasti surinkę komandą:

`$ man g++`

### **3. Iš daugelio failų sudarytų programų kompiliavimas**

Iš daugelio failų sudarytoms programoms yra naudojama pagalbinė programa *make*.

Prieš naudojant *make*, reikia sukurti failą *makefile*, kuriame tam tikru formatu ("taisyklėmis") reikia aprašyti ryšius tarp failų jūsų programoje.

*makefile* failo tipinė struktūra yra tokia:

*tikslas: prielaidos ...*  
*komanda*

...

...

*Tikslas* – paprastai kompiliatoriaus sukuriamos programos vardas, pvz., objektinio modulio arba vykdomojo failo vardas.

*Prielaida* – failas, kuris yra naudojamas sukuriant tikslo failą. Paprastai tikslo failui sukurti reikalingi keli failai.

*Komanda* – veiksmas, kurį turi atlikti *make* programa.

Failu *makefile* gali būti aprašyta daug tokių taisyklių. Kiekviena taisyklė yra vykdoma tuomet, kai pasikeičia bent viena prielaida. Taisyklės, kurios vykdo tik komandas, gali neturėti prielaidų.

Pabandykime sukurti 2 pavyzdžio taisyklių failą *make* programai:

```
# failas makefile

# sukurti vykdomąjį failą calculate
calculate : calculate.o suma.o
    g++ -o calculate calculate.o suma.o

# sukompiliuoti failą calculate.cpp
calculate.o : calculate.cpp suma.h
    g++ -c calculate.cpp

# sukompiliuoti failą suma.cpp
suma.o : suma.cpp suma.h
    g++ -c suma.cpp

# ištrinti laikinus failus
clean :
    rm calculate.o suma.o
```

Šis failas aprašo auomatizuotą programinės sistemos sudarytos iš failų *calculate.cpp*, *suma.cpp* ir *suma.h* kompiliavimą.

Norint naudoti šį failą tereikia įvykdyti komandą *make*:

`$ make`

Sukuriamas vykdomasis failas *calculate*.

Plačiau apie *make* programą galite paskaityti įvedę komandą:

`$ man make`