

Arenduskeskonna ja -vahendite seadistamine

Tekstiredaktor **Gedit**

Edit > Preferences

View tab:

- Display line numbers ☒
- Display margin at column
- Enable text wrapping ☐
- Highlight current line ☒
- Highlight matching brackets ☒

Editor tab:

- Tab with
- Insert spaces instead of tabs ☒
- Enable automatic indentation ☒

View > Highlight mode select C (või vali alumiselt menüüribalt).

Gedit pistikprogrammid (pluginad)

C koodi redigeerimiseks kasulikud pluginad (vaja paigaldada: `gedit-plugins gedit-developer-plugins gedit-source-code-browser-plugin`):

- `Code comment`. Koodi kommenteerimine ja kommentaari lisamine parema hiireklahvi rippmenüüst
- `Draw spaces`. Tühikute ja reavahetuste kuvamine
- `Embedded Terminal`. Terminali tab gedit aknas
- `External Tools`. Välise programmide ja skriptide käivitamine.
- `Find in Files`. Otsimine kõikidest kataloogi failidest.
- `Git`. Muudetud koodi näitamine.
- `Smart Spaces`. Liigenduse tagasivõtmine nii nagu kasutaks tabulaatorit
- `Source Code Browser`. Lähtekoodi "Side bar"
- `Word Completion`. Sõnade automaatne lõpetamine

Lülita sisse ja seadista Edit > Preferences > Plugins tab.

Kasulikud klahvikombinatsioonid

Shift+Ctrl+F	Find in files
Ctrl+F8	Build
Alt+F12	Remove trailing spaces
Ctrl+I	Goto line

Lähtekoodi vormindamise tööriistad

Clang-Format

Paindlik käsureatööriskoodi vormindamiseks [YAML formaadis konfiguratsioonifailide](#) alusel.

Kahjuks ei toeta vaikumisi K&R stiili.

Konfiguratsiooni faili koostamisel saab kasutada [clang-format](#) veebitööriista näiteid.

Artistic Style

Kiire ja väike automaatne koodivormindaja.

Lähtekoodi haldur Git

- Seadista enda nimi ja e-post

```
$ git config --global user.name "Name"
$ git config --global user.email "email"
```

- Vajadusel seadista üldine `.gitignore_global`. Vaata [GitHub gitignore põhju](#) oma operatsioonisüsteemi ja tööriistade järgi. Lisa vajalik oma kodukataloogi faili nimega `.gitignore_global`. Ära unusta seda seadistustesse lisada:

```
$ git config --global core.excludesfile '~/gitignore_global'
```

- Kui kasutad Windows operatsioonisüsteemi, siis taga et reavahetused oleks "Unix" stiilis. [GitHub help - Dealing with line endings](#).

Tarkvaraprojekti ja repositooriumi loomine

NB! Kuna tarkvaraprojekt on nii päevasel kui ka kaugõppe kursusel ühine siis on seal omajagu üleliigset mida teil esimestes laborites vaja ei lähe.

- Klooni enda `Documents` (See kataloog on Kolledži tööjaamades üle võrgu kättesaadav) kataloogi skelett:

```
$ git clone https://gitlab.com/I237-2016/skeleton.git
```

- Nimeta kataloog enda soovi järgi ümber, näiteks:

```
$ mv skeleton my-course-repository
```

- Nimeta skeleton repository ümber:

```
$ cd my-course-repository/  
$ git remote rename origin skeleton
```

- Loo GitLabis või GitHubis uus projekt. (Näites kasutatud <https://gitlab.com/I237-2016/my-course-repository.git>)

Lisa loodud projekt kui uus origin:

```
$ git remote add origin https://gitlab.com/I237-2016/my-course-repository.git
```

Selline lähenemine võimaldab uuendada projekti skeletti ilma käsitsi faile kopeerimata.

```
$ git pull skeleton master
```

Loe lisa [Git Basics Working with Remotes](#).

AVR "tööriistakast"

- `avr-gcc` - translaator(compiler) ja linker. [GNU Compiler Collection](#). `avr-gcc` spetsiifilised võtmed. GCC 4.9.2 dokumentatsioon. [avr-gcc Wiki](#)
- `avr-objcopy` - objektfailide translaator. [man internetis](#) või `man objcopy`.
- `avrdude` - ehitatud koodi paigaldamise programm. [AVRDUDE user manual](#)
- `avr-libc` - AVR mikrokontrollerite C teek. [The C runtime library for the AVR family of microcontrollers](#) **NB!** projekti lehel on versioon 2.0.0 dokumentatsioon. Klassi arvutites on versioon 1.8. Erinevused on väiksed aga need on olemas. Paigaldatud versiooni dokumentatsioon asub `/usr/share/doc/avr-libc/avr-libc-user-manual`