

Аудиториски вежби 1

Вовед
Околина за развој



М-р Ѓорѓи Маџаров
М-р Томче Делев

Структурирано програмирање

Факултет за информатички
науки и компјутерско
инженерство - Скопје 2011

Содржина

Вовед во околини за програмирање

Code::Blocks - инсталација



Програмирање (1)

- Програмите, што компјутерот ги извршува се последователност од нули и единици, затоа што тоа е единствениот јазик кој што компјутерот го разбира
- Програмерите ги пишуваат своите програми на јазици за програмирање, кои што се разбирливи за нив
- Програма напишана во јазик за програмирање ја нарекуваме **изворна програма**



Програмирање (2)

- За пишување програми често се користат **околини за развој**
- Програмата се внесува преку текстуален уредувач
- Потоа се врши преведување на програмата (компајлирање)
- Со тоа се создава извршна програма т.е. програма напишана во јазикот на компјутерот



Елементи на околните за развој

Околината за развој е составена од повеќе програми, кои го олеснуваат целокупниот развој на една програма

- текст уредувач (text editor)
- преведувач (компајлер - compiler)
- дебагер (debugger)
- интеграција на библиотеки со функции
- поврзувач (linker)



Текст уредувач (text editor)

- Програма која овозможува внесување и уредување на текстот на изворната програма
- Овозможува зачувување на програми и вчитување на веќе напишани програми за нивно повторно уредување
- Означување на клучните зборови и команди во изворната програма (syntax highlighting)



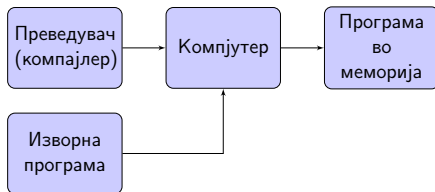
Текст уредувач (text editor)

- Ја преобразува (преведува) изворната програма од јазикот за програмирање во кој е напишана во јазик разбирлив за компјутерот
- Се разликуваат два вида преведувачи: **интерпретери** и **компајлери**
- Интерпретер е преведувач кој ја *обработува одделно секоја команда*, ја проверува за грешки и ја извршува, по што поминува на следната команда итн.
- Компајлер е преведувач кој ја *обработува целата програма*, ја проверува за грешки и ја преведува, по што се добива извршната програма.
 - Така добиената извршна програма може да се извршува

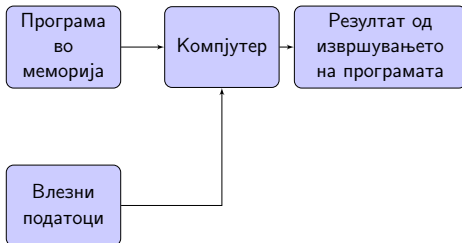


Тек на преведување и извршување на програма

Фаза 1 - преведување на програмата



Фаза 2 - извршување на програмата



Дебагер (debugger)

- Компајлерите и интерпретерите ги откриваат грешките (синтаксички) во програмата поради не правилно користење на јазикот за програмирање
- Друг вид на грешки се логичките грешки
 - Програмата не го прави тоа за кое што е наменета
 - Се откриваат многу тешко
- Дебагер е програма која помага при барање на логичките грешки
 - Овозможува следење на извршувањето на програмата чекор по чекор



Интеграција на библиотеки со функции

- Интегрирање и користење на претходно создадени и проверени модули (потпрограми), уште наречени и функции
- Ваквиот начин на организација на програмите има голем број на предности
- Повторно искористување на готови функционалности
- Пример библиотеки
 - За управување со стандардниот влез и излез
 - За стандардни математички операции



Поврзувач (linker)

- Понекогаш програмата е премногу голема за да се напише во една датотека
 - различните делови може да се пишуваат од различни програмери.
 - некои делови од дадена програма можат да бидат искористени и во друга програма
 - Одделно компајлираните делови е неопходно да бидат обединети во една цела извршна програма со помош на **поврзувачот**
 - Друга улога на поврзувачот е да ги „сврзе“ со програмата потребните библиотеки со стандардните функции



Околини за развој

(ntegrated Development Environment - IDE

- Сите овие елементи на околината за развој се обединуваат (интегрираат) во т.н. интегрирани околини за развој
- Пример за IDE е околината која ќе се користи на овој курс, Code::Blocks



Содржина

Вовед во околини за програмирање

Code::Blocks - инсталација



Code::Blocks - инсталација

- Како да го најдеме и инсталираме Code::Blocks
- Code::Blocks е **слободен софтвер** и може да се најде на <http://www.codeblocks.org/downloads>
- Во централниот дел на страната има три линка: **Download the binary release**, Download the source code и Retrieve source code from SVN
- За наједноставна инсталација се препорачува да се избере првиот линк - **Download the binary release**,

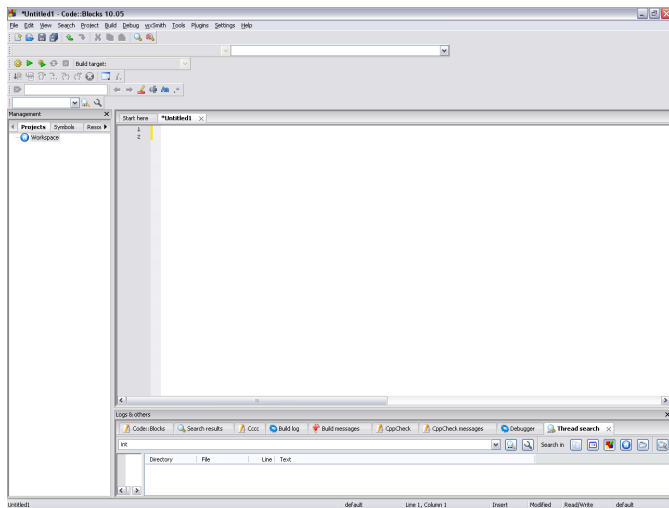


Code::Blocks – инсталација (2)

- За почетниците се препорачува да ја симнат верзијата што во нејзе вклучува **MinGW** setup
 - моментално тоа е линкот **codeblocks-10.05mingw-setup.exe** кој е наменет за корисниците на сите **Windows** оперативни системи
 - Со клик на изворот Sourceforge.net се отвора нова страницата која по истекот на 5 секунди сама ќе ви понуди опција да ја зачувате датотеката на од вас избрана локација
 - По зачувувањето на датотеката следете ги инструкциите за инсталирање



Code::Blocks – главен прозорец



Елементи на главниот прозорец

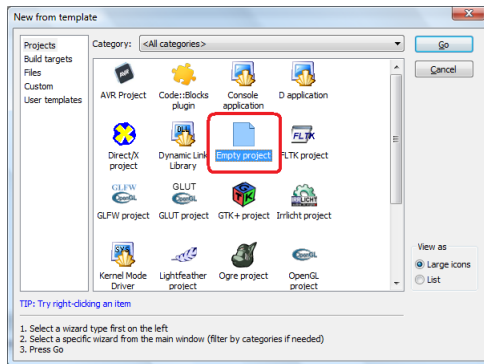
- Лента со менија
 - лентата со менија се наоѓа во најгорниот дел на прозорецот, веднаш под неговиот насловот
 - Во неа се наоѓаат менијата File, Edit, View, Search, Project, Build, Debug, wxSmith, Tools, Plugins, Settings, Help
- Лента со алатки
 - лентите со алатки (копчиња за стартување на најчесто користените команди на околината) се наоѓаат непосредно под лентата со паѓачки менија
- Работна површина
 - Потпрозорец за уредувачот на текст
 - Прозорец за соопштенија.
 - Прозорец за организација на работата на програмата



Програмирање во C со Code::Blocks

Креирање проект

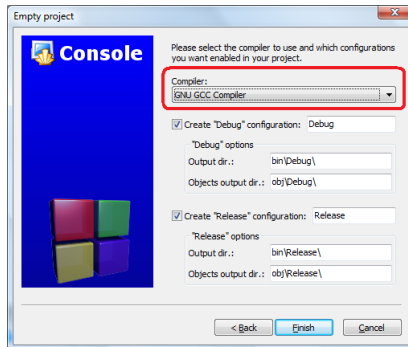
1. Стартувајте CodeBlocks
2. File -> New -> Project -> Empty Project -> Go



Програмирање во C со Code::Blocks

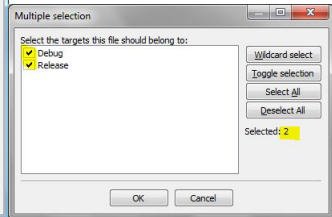
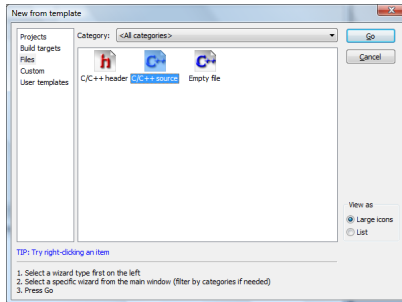
Креирање проект

3. Одберете GNU GCC Compiler
4. Изберете ги следните 2 опции ако сакате да креирате “debug” и “release” configuration



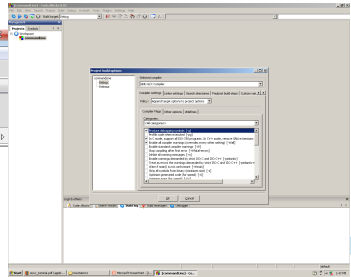
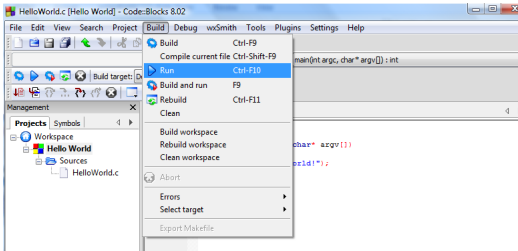
Додавање на изворна датотека

5. Додадете изворна датотека во проектот: File -> New -> File -> C/C++ Source
6. Одберете C како програмски јазик
7. Внесете го името на датотеката со полната патека и не заоравајте да го вклучите "Add file to active project"



Програмирање

- За секој проект може да се постават следните опции "Project Build Options.. Compiler Flags"
- За изградба на проектот (build) притиснете Ctrl + F9
- За извршување на проектот притиснете Ctrl + F10



Задачи за дома

- Во продолжение се наведени неколку задачи кои би требало да се обидете да ги изработите дома
- Со нивна изработка ќе бидете подготвени за успешна работа на претстојните лабораториски вежби



Задача 1

Обидете се да креирате нов проект со една .c датотека и во неа внесете го текстот на следнава програма:

```
#include <stdio.h>

int main() {
    printf("Zdravo, kako si?\n");
    return 0;
}
```



Задача 1

- Извршете ја програмата
 - Што добивате како резултат?
- Доколку сте направиле грешка при пишувањето на текстот поправете и извршете уште еднаш.
- Направете намерно некоја грешка во текстот. Извршете повторно!
 - Што се случува сега?



Задача 2

Во текстот на програмата додадете до означениот ред:

```
#include <stdio.h>
int main() {
    printf("Zdravo, kako si?\n");
    printf("Neshto ne ti se pravi muabet?");
    return 0;
}
```

Кој е резултатот од извршувањето сега?



Материјали и прашања

Предавања, аудиториски вежби, соопштенија
courses.finki.ukim.mk

Изворен код на сите примери и задачи
bitbucket.org/tdelev/finki-sp

Прашања и одговори
qa.finki.ukim.mk

