

Organisation of source code

Organisation of source code:

- Files ending in `.c` contain code for functions and procedures;
- Files ending in `.h` contain interfaces to API's (definitions of variables, functions and procedures);

API's called *libraries* in C

A file ending in `.h` is called an *include-file*

⇒ at start of each source code file, include all necessary interfaces to used API's (line 1 in the hello-World example).

Name of executable fixed at compile-time: default is name of program containing `main`-function.



The program development cycle

Suggest to use eclipse-IDE. Details are on the web.

Command-line based approach:

- Create source code with editors (emacs, gedit). Will use emacs in this module.
- Compile code. The command for this is `cc -Wall -Werror -o <filename> <filename>.c`. Best started from within emacs from the menu via `Tools->Compile` or typing `M-x compile`.
- Use the emacs-command `C-x'` to locate all errors.

Execute code: Open shell and execute command by typing its name, prefixed by `./`.



Example 1

Always the same: Print "Hello World!"

```
#include <stdio.h>

int main () {
    printf ("Hello World\n");
    return 0;
}
```



The standard C-Library

Comprehensive collection of functions for low-level system tasks (including input/output) and strings

Explanation of functions also available via the command-line:

```
man -S 3 printf
```

will explain the `printf`-function



How to find missing include-files

Compiler will produce warning messages or errors if include-files are missing

manual pages list necessary include-files at the top

Example: `stdlib.h` is necessary for `exit` in `checknum.c`



Tools for debugging

Eclipse has good support for this—see webpages.

Stand-alone debuggers also exist (eg `ddd`)

Important: If debugger is to be used, program needs to be

compiled with `-g` option, eg

```
cc -Wall -Werror -g -o helloWorld helloWorld.c
```

Done automatically by IDE



Basic data types in C

Basic types are `int`, `char`, `float` and `double`.

`char` is subset of `int` containing valid character codes

Booleans not officially standard, but standard definition has emerged:

```
typedef int boolean;
#define TRUE 1
#define FALSE 0
```

Note: `TRUE` must be $\neq 0$ and `FALSE` must be 0 or if-statements to work;

Note: *Strings not basic datatype!*

Will see later how to encode them.



Commandline parameters

`main`-function accepts arguments:

- `int argc` stores number of parameters
- `string argv[]` stores the parameters as an array of strings (need to define type of strings in the program for this to work)



Generating Output

Output on the terminal generated by `printf`-procedure.

First argument is *format string*: Output is this string, with sequence of substrings “%c” where *c* is a character, replaced by the remaining arguments, converted to a string.

Type of arguments specified as follows:

Format character	Type of argument
%c	char
%s	string
%d	int