Metropolia University of Applied Sciences

Programming

TI00AA43-3002

Lecture 5

Sami Sainio

sami.sainio@metropolia.fi

Plan for going forward

- Variables and printing to screen
- If, else, while and for loops
- I/O
- Functions and tables
- **File handling**
- Pointers and arrays
- Simple structures
- Program structure and design

5. Lectures

File handling

File handling

- Program code that reads files is very useful when necessary to have large files that are not necessarily wanted to be in the RAM memory during the whole execution / handling of the data
- Communication between different software is possible when using files as well (for example log-file readers etc.)

File handling

- To open a file it is necessary to point to the memory location where the file is located at
- FILE *filePointer; // creates pointer
- Now the pointer still needs a location where it is pointing at. This is accomplished by calling function that is used for opening the file: fopen()

Opening file

- fopen() needs two parameters. 1st is the file location, 2nd the mode that we want to open the file in

```
FILE *filePointer;

filePointer= fopen(”c:\\test.txt”, ”r”);    // open
fclose(filePointer);                        // close
```

- Now filePointer is pointing to the path given to a file ”test.txt” and it is opened for reading (r)
- NOTE! Programmer must close the file!

Opening file

- Modes where file can be opened

r	Read
w	Write (file doesn’t have to exist)
a	Append (file doesn’t have to exist)
r+	Open for reading, start from beginning of the file
w+	Overwrite file if exists
a+	Append if file exists

File I/O

- File write can be done among with other functions:
  - fprintf(), fputc()
- File reading:
  - fgetc()
- fputc() and fgetc() work one character at a time and remind us from getchar() and putchar() functions and work very similarily

Reading file one character at a time

```
FILE *filepointer;

filepointer = fopen("c:\\testi\\test.txt", ”r");

char mark;

while((mark= fgetc(filepointer)) != EOF)
{
        printf(”character read: %c”, mark);
}

fclose(filepointer);
```

Writing to file one character at a time

```
FILE *filepointer;

filepointer = fopen("c:\\testi\\test.txt", ”w+");

char mark;

/* read user input with for example getchar() function in while loop untill we receive */
/* terminating character, for example !   */

            fputc(mark, filepointer);


fclose(filepointer);
```