T-409-TSAM-2016: Computer Networks
Programming Assignment 0 – Propaedeutics

Lecturer: Marcel Kyas

August 17, 2016

Submission Deadline

The solution will be discussed in your lab session on 23.08.2016. The solution need not be submitted and will not be graded. You may submit the solutions in myschool and ask for feedback.

Intended Learning Outcomes

You should be able to:

- Work with `gcc` and `make`.
- Write basic programs in the C program language.
- Get to know the C standard library and the X/Open library.

1 Instructions

The programming exercises for this course will be delivered in the C programming language (Kernighan and Ritchie, 1988). More specifically, we will use ISO C11, as standardized in ISO/IEC 9899:2011.

I assume that everybody can write simple programs in C and C++, because these skills are learning outcomes of T-107-TOLH, Tölvuhögun (“Write and debug simple programs in the C programming language”) and T-201-GSKI, Gagnaskipan.

This is a propaedeutics sheet, you can catch up on the things you need for network programming. You get acquainted with programming C, a Unix tool chain, and can practice some elements that may not have been taught before.

2 Hello, world!

The classical first program is one that prints “Hello, world!” If you get this one running, you operate the tool chain successfully.

The C programming language has been revised multiple times, getting more clear with each update. Invoking `gcc` with the flag `-std=c11` means that the source code language conforms to C11,[1]

Since the C programming language deliberately leaves a lot unspecified, we have to specify the environment for that the source code is written. Otherwise, functions provided by the operating system may not be made available. The common way to do this is to set feature test macros. Setting the feature test macro `_XOPEN_SOURCE` to `700` indicates source code written for an environment that requires the X/Open System Interface (XSI), version 7 or later, a superset of Portable Operating System Interface (POSIX).

You can study and modify the file `Makefile.template`. The program `make` is a build system used on UNIX systems. It is documented at `https://www.gnu.org/software/make/manual/html_node/index.html`.

1. Compile and execute the program in the file `hello.c`, displayed in Listing 2.1.

Listing 2.1: Hello, world

```
#include <stdio.h>

int main()
{
    printf("Hello, world!\n");
    return 0;
}
```

3 Arrays and Pointers

C11 has pointers and arrays as data types. A pointer is not an array, but pointers can be indexed as arrays. But arrays are different from pointers.

2. Compile and execute the program in the file `arrays.c`, displayed in Listing 3.1, and explain its output.

[1] Prior to gcc version 4.9, support for C11 is incomplete and experimental. The features needed for this course should be working with gcc since version 4.7.

Listing 3.1: Arrays listing

```c
#include <stdio.h>

int a[10];

int main()
{
    printf("%p\n", a);
    printf("%p\n", &a[0]);
    printf("%p\n", &a);
    printf("%zu\n", sizeof(a));
    printf("%zu\n", sizeof(&a));
    printf("%zu\n", sizeof(a[0]));
    printf("%zu\n", sizeof(&a[0]));
    printf("%zu\n", sizeof(*(a)));
    printf("%zu\n", sizeof(*(&a)));
    printf("%zu\n", sizeof(*(&a[0])));

    return 0;
}
```

4 Strings

The C programming language does have a designated data type for strings and does not specify functions that operate on strings of texts as language primitives. Strings are considered a library issue, which allows maximal flexibility.

The standard library specifies the header `<string.h>`. Strings are considered as sequences of `char`, which is commonly interpreted as a character, terminated by NUL `'\0'`. Formally, `char` is an integral type that has at least 8 bits. A variable of type `char` can hold any integral value in the interval `CHAR_MIN` to `CHAR_MAX`, as specified in the header file `<limits.h>`. The values are usually interpreted as American Standard Code for Information Interchange (ASCII) (or Unicode Transformation Format (UTF)-8) encoded characters.

The POSIX additionally provides the headers `<wchar.h>` and `<wctype.h>`. These headers specify an application programming interface (API) that works on 8-bit and multi-byte encodings like UTF-8, Big5, Shift Japanese Industrial Standards (SJIS), and others.

The encoding of strings in computer networking applications depends on the protocol. For this sheet, assume standard library strings.

3. Write a program to print a histogram of the lengths of words in its input. Read from standard input until the end of the input file. You may use `getc()` and `ctype.h` macros.[2] It is easy to draw the histogram with the bars horizontal; a vertical orientation is more challenging.

[2] On UNIX, the command `man` can provide you with documentation and example code. `man getc` displays documentation about the `getc()` function.

5 Files

C11 using the XSI has two distinct API for working with files. The functions defined in the C standard library are defined in the header file `<stdio.h>` and use the functions `fopen()`, `fclose()`, `fread()`, and `fwrite()`. POSIX defines the functions `open()`, `close()`, `read()`, and `write()` (see the documentation for the required headers). The first interface provides buffering and other features and is often implemented using the second. The second interface is usually implemented by the operating system. The first interface is preferred if portability is a concern.

4. Write a program to compare two files, printing the first line where they differ.

Acronyms

API application programming interface.

ASCII American Standard Code for Information Interchange.

HTTP Hypertext Transfer Protocol.

POSIX Portable Operating System Interface.

SJIS Shift Japanese Industrial Standards.

SMTP Simple Mail Transfer Protocol.

UTF Unicode Transformation Format.

XSI X/Open System Interface.

References

ISO/IEC 9899:2011. *Information technology – Programming languages – C*. International Organization for Standardization (ISO), Geneva, Switzerland, 2011.

Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1988.