ELTE FI, Computer Science BSC
Operating Systems
http://opsystems.inf.elte.hu

Responsible lecturer: Illés, Zoltán associate professor
Media & Educational Informatics Department

1. Introduction

Goal: We are getting to know our server: how to log in, how to compile files and how to execute them. We learn a bit about C language as well, the difference between C++ and C.

We learn/refresh*: cc or gcc fname –o outputfile –Wall – compiling; gdb – debugger; chmod – change permissions, ./name – execution; #include file – include library files; arguments of main function; printf, scanf, fgets – read, write; strcat, strcpy, strlen – string functions (there is no string type!); malloc, calloc, realloc, free – memory allocation; pointers – instead of reference parameters*

Tasks

1. Log in to the server **opsystems.inf.elte.hu** (using scp and ssh)! The username is your Neptun-code, the password is given by your teacher!

2. Write a hello world program in C!
   You can use an editor on the server (vi, joe, mcedit) or use notepad++ on your Windows client and transfer it via winscp. Be careful, the file extension of the code must be .c! On the server (you are logged in by telnet, ssh) you have to compile it with gcc/cc. Check whether it has an execcution permission or not – if not, give using chmod u+x. Execute it by typing ./name!

   ```
   #include <stdio.h>
   #include <stdlib.h>
   int main(){
     printf(„hello world\n”);
    return 0;
   }
   ```

3. Write a C program which declares a char array variable and allocate 80 bytes for it. (*char word[80] or char* word=(char*)malloc(80);*) Read a word from the keyboard with scanf/fgets functions into the variable. Decide the length of it with strlen built in function! (*Do not forget about including string.h! Is there any difference between the usage of scanf and fgets? How can you read in more then one word?)*

4. Modify the above written program and write an own length function (instead of strlen)! *(You must remember that the end of a string is signed by a 0 value – so iterate till you do not get a 0!)*

5. Write a function (in a C program) which gets a char array as an argument and gives back the same „string” without the starting spaces! E.g. „   apple” -> „apple” *(Please, iterate through the spaces and give back the pointer which shows to the first real character (not space)!)*

6. Write a C program which gets values as arguments from the command line and writes them out on to the screen! (*Use parameters in the main function! – int argc, char** args. In argc there is the number of arguments from the command line. Char** - is an array of char arrays – so it is a „string” array)*

7. Write a C program which gets a number N from the command line and write out „hello” N times! (*You should remember that you get the number as a char array so you have to convert it to int using atoi function!)*

8. Write a function which can modify the content of a „string” array so that you can replace word1 to word2 at each occurence! (*Use strcpy function in string.h! What happens if word2 is longer than word1? Why?)*

Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government