Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Fall Term 2016

SYSTEMS PROGRAMMING AND COMPUTER ARCHITECTURE
Assignment 2: Introduction to C Programming

Assigned on: **29th Sep 2016**
Due by: **6st Oct 2016**

Introduction to C Programming

This exercise consists of small C programming problems.

1 Reverse an array

Write a C program that has a function that:

- accepts an array of 32-bit unsigned integers and a length
- reverses the elements of the array in place
- returns void (nothing)

Example for an array of length 4: $\boxed{5}\ \boxed{2}\ \boxed{7}\ \boxed{5} \rightarrow \boxed{5}\ \boxed{7}\ \boxed{2}\ \boxed{5}$

2 First whitespace-separated word

Write a C program that has a function that:

- accepts a string as a parameter
- returns the first whitespace-separated word in the string (as a newly allocated string) and the size of that word

Example: “lorem ipsum” → (“lorem”, 5)

3 Box-and-arrow diagram

Use a box-and-arrow diagram for the following program to explain what it prints out:

```
#include <stdio.h>

int foo(int *bar, int **baz)
{
  *bar = 5;
  *(bar+1) = 6;
  *baz = bar+2;
  return *((*baz)+1);
}

int main(int argc, char **argv)
{
  int arr[4] = {1, 2, 3, 4};
  int *ptr;
  arr[0] = foo(&(arr[0]), &ptr);
  printf("%d %d %d %d %d\n",
  arr[0], arr[1], arr[2], arr[3], *ptr);
  return 0;
}
```

4 Little vs. big endian

Write a C program that prints out whether the computer it is running on is little endian or big endian. (hint: pointer and casts)

5 Binary search tree

Implement and test a binary search tree[1] in C:

- implement key insert() and lookup() functions
- implement it as a C module: bst.c, bst.h
- implement test_bst.c (contains main(), tests out your BST)
- don't worry about making it balanced
- as a bonus implement a key delete() function

[1] http://en.wikipedia.org/wiki/Binary_search_tree

6 Function pointers basics

Write a C program that has a function that:

- accepts a function pointer (pointing to a function with an integer return type and a single integer argument) and an additional array of integers and length of the array as arguments
- invokes the pointed-to function with each of the elements in the array as an argument
- overrides the current array element with the return value of the called function

Example: The function `comp` provided as a function pointer along with the array [-1] [3] [-27] should yield [0] [1] [0].

```
int comp(int a)
{
    if (a <= 0) return 0;
    else return 1;
}
```

Hand In Instructions

Question 4 is a pen-and-paper exercise. Hand it in to your assistant during the exercise session. For the rest of the problems, upload your source files to a subfolder named **assignment2** in your SVN folder. Refer to Assignment 1 for instructions on using SVN.