ICSI333 System Fundamentals

Note: Students are expected to start the activities as soon as the description is available and seek feedback as needed. Although some activities are not graded for credit, they are contiguous study of the lecture or used as stepping-stones for the projects. Skipping any activities would impact the learning significantly.

Objectives:

- Practice pointers

Reading:

- Lecture notes

Submission(5 points):

- All required C programs must be submitted on Duifene on time.

Instructions:

Task #1. Boolean type in C

Without executing the following program, determine the output.

```
#include<stdio.h>
int main () {
  int i = 5;
      if(i == i == i)
            printf("Yes\n");
      else
            printf("No\n");
  return 0;
}
```

The output is "No". The Boolean expression `5 == 5 == 5` is evaluated from left to right. The first == (`5 == 5`) returns true, which is `1` in C. And then, the second `== (1 == 5)` returns false, which is `0` in C. Hence, "No" is printed. Add the following statements in the program and verify the results.

```
printf("%d\n", (i == i));
printf("%d\n", (i == i == i));
```

Change the value of variable `I` to 1 (one) and run the program again.

Task #2. Pointers

Study the following example. Explain changes made after each executable statement.

```
#include <stdio.h>

int main(){
      int x = 10;
      int *ptr;

      ptr = &x;
      printf("%p\n", ptr);
      printf("%p\n", &x);
      printf("%d\n", *ptr);
      printf("%p\n", &ptr);
```

```
        *ptr = 20;
        printf("%d\n", x);

        x = 30;
        printf("%d\n", *ptr);

        return 0;
}
```

Task #3. Create a program that uses pointers

The program must declare two integer variables and a pointer to each of the two pointers. Set the first integer variable to a value, dereference the second pointer to change its value and print all four values.

Task #4. Explain why the following program produce different results at each run

```
#include <stdio.h>

int main(){
        int x;
        printf("%p", &x);

        return 0;
}
```

Task #5. Pointer Expressions and Pointer Arithmetic

A limited set of arithmetic operations can be performed on pointers. A pointer may be incremented (++) or decremented (--); an integer may be added to a pointer (+ or +=) and subtracted from a pointer (– or -=). Pointer arithmetic is meaningless unless performed on an array. Study the following example and explain the logic.

```
#include <stdio.h>

int main(){
        int v[3] = {10, 100, 200};
        int *ptr;

        // Assign the address of v[0] to ptr
        ptr = v;
        printf("Value of *ptr = %d\n", ptr[0]);
        printf("Value of *ptr = %d\n", ptr[1]);
        printf("Value of *ptr = %d\n", ptr[2]);

        for (int i = 0; i < 3; i++){
                printf("Value of *ptr = %d\n", *ptr);
                printf("Value of ptr = %p\n\n", ptr);
                ptr++; // Increment pointer ptr by 1
        }
}
```

Linux Basic Commands

`ls [option(s)] [file(s)]`	If you run ls without any additional parameters, the program will list the contents of the current directory in short form. `-l` detailed list `-a` displays hidden files
`cp [option(s)] sourcefile targetfile`	Copies sourcefile to targetfile. `-I` waits for confirmation, if necessary, before an existing targetfile is overwritten `-r` copies recursively (includes subdirectories)
`mv [option(s)] sourcefile targetfile`	Copies sourcefile to targetfile then deletes the original sourcefile. `-b` creates a backup copy of the sourcefile before moving. `-I` waits for confirmation, if necessary, before an existing targetfile is overwritten.
`rm [option(s)] file(s)`	Removes the specified files from the file system. Directories are not removed by rm unless the option `-r` is used. `-r` deletes any existing subdirectories `-I` waits for confirmation before deleting each file
`cd [options(s)] [directory]`	Changes the current directory. cd without any parameters changes to the user's home directory.
`mkdir [option(s)] directoryname`	Creates a new directory.
`rmdir [option(s)] directoryname`	Deletes the specified directory, provided it is already empty.
`cat [option(s)] file(s)`	The cat command displays the contents of a file, printing the entire contents to the screen without interruption. `-n` numbers the output on the left margin
`cal`	Displays the calendar of the current month.
`date`	Displays current time and date.
`whoami`	Reveals the user who is currently logged in.
`whatis`	Gives a one-line description about the command. It can be used as a quick reference for any command.

`man`	Manual Pages, for more detailed information, Linux provides man pages and info pages. To see a command's manual page, man command is used.
`pwd`	Prints the absolute path to current working directory.
`vi`	A text editor for Linux operating system.
`gedit`	The default GUI text editor in the Ubuntu operating system.
`emacs`	Another text editor for Linux operating system.

Vi commands:

To save and quit:

Commands	Action
:wq	Save and quit
:w	Save
:q	Quit
:w fname	Save as fname
ZZ	Save and quit
:q!	Quit discarding changes made
:w!	Save (and write to non-writable file)

More commands:

Copy-pasting within the terminal: Ctrl+Shift+C/V

Resources and Credits:

- Teaching materials from Professor Kuperman at UAlbany
- Vi reference: https://www.ele.uri.edu/faculty/vetter/Other-stuff/vi/vi-quick-ref.pdf