

CS101 Lab-2

Today's Linux Commands

Recall how to open terminal and write commands from the previous tutorial. Recall that `.`, `..`, and `~` represents the current directory, parent directory and home directory respectively.

- `mkdir -p ~/lab2g1` tells `mkdir` to create all necessary directories if they don't already exist
- `cd ~/lab2g1` to change your working directory to `~/lab2g1`
- Recall how to write, compile, and run C programs from previous labs.
- Learn the following linux commands while you work with today's programs
 - a. `*` is a special character that matches with any number of any character except `/`
 - b. Write `ls *.c` to list all `.c` files
 - c. `mkdir program_copies` to create a new directory
 - d. `cp *.c program_copies` to copy all `.c` files to new directory
 - e. `ls program_copies` to list all files in the new directory
 - f. **(use carefully!)** `rm program_copies/*a*.c` to remove all `.c` files (if exist) that contain character `a` in their name. Note that there is no concept of 'Recycle Bin' for the terminal, deleted files are gone forever!
 - g. `rmdir program_copies` will show error since `program_copies` is not empty
 - h. `rm program_copies/*.c` to delete all files and `rmdir program_copies` to finally remove directory
 - i. You could also use `rm -r program_copies` to recursively delete all files in the directory `program_copies` and then the directory `program_copies` itself
 - j. Use `cat *.c` to see all `.c` program contents on terminal
 - k. Printing executable files directly to the terminal (e.g. `cat hello.out`) prints garbage. To see the file content in hex, use `hexdump hello.out`

Instructions for the lab assignment:

When you are done with the following programs, raise your hand and a **lab instructor will verify your second program and ask a couple of questions**. You will be graded upon the correctness of your program, naming of your variables, and indentation and comments in your code.

1. Write a C program to take input of two signed integers, say, **a** and **b**. It should then correctly print **a+b a-b a*b a%b a/b**(integer division) **a/b**(real division). Typecast your variables correctly to make this happen.
2. Write a C program to determine the precedence order and the direction of evaluation for the following operators:
 - a. unary **-**
 - b. assignment **=**
 - c. typecast (e.g. **float, int**)
 - d. arithmetic **+ - * / %**

After you write your program, write down a precedence table as discussed in the lecture and show it to the lab instructor.

3. Write a C program that takes input two integers **a** and **b**. It should then swap their values and print the swapped values. That is, write lines of code in the following program to make that happen.

Hint: Use a new variable to hold a value.

Bonus challenge: Assume the integers are not too big, say, both are **<1e8** in absolute values. Swap their values without allocating fresh memory.

```
#include<stdio.h>
int main() {
    int a, b;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);
    printf("a=%d and b=%d before swap\n", a, b);
    /*
        Write your code here
    */
    printf("a=%d and b=%d after swap\n", a, b);
    return 0;
}
```

4. Type and run the following two programs.

```

#include <stdio.h>
int main() {
    short short_x = 32767; //Signed--uses 15 bits
    int int_x=-65; //Signed--uses 31 bits
    unsigned long long_x=400000; //Unsigned--uses 32 bits

    printf("short_x: Decimal:%hd, Octal:%ho, Hex:%hx\n", short_x, short_x, short_x);
    printf("int_x: %d\n", int_x);
    printf("long_x = %lu\n", long_x);

    /* Data too large for type --causes overflow*/
    short_x = short_x + 1;
    /*16-bit value can't be held in 15 bits*/
    printf("short_x after incrementing: %hd\n", short_x);

    /*Wrong selection of format specifier*/
    printf("int_x printed with %%u = %u\n", int_x);
    int_x=123456;
    printf("int_x printed with %%d %d, with %%hd= %hd\n", int_x, int_x);
    return 0;
}

```

```

#include <stdio.h>
int main(){
    float pi1 = 3.142, pi2 = 0.3142E1, pi3 = 3142E-3;
    printf("pi1=%f, pi2=%e, pi3=%g\n", pi1, pi2, pi3);

    float float_x = 12345.678901234567890; //20 significant digits
    double double_y = 12345.678901234567890; //20 significant digits
    printf("float_x = %.15f\n", float_x);
    printf("double_y = %.15f\n", double_y);

    long double ldouble_z = 12345.678901234567890;
    printf("ldouble_z = %.20LF\n", ldouble_z);

    float sensex1 = 81904.70;
    float sensex2, gain;
    printf("Enter today's value of Sensex: ");
    scanf("%f", &sensex2);
    gain = (sensex2 - sensex1) / sensex1 * 100;
    printf("Sensex change since last close: %0.2f%%\n", gain);
    return 0;
}

```