Intro to C programming and Ubuntu Commands

Course: Introduction to Programming and Data Structures

Laltu Sardar

Institute for Advancing Intelligence (IAI),
TCG Centres for Research and Education in Science and Technology (TCG Crest)

Januray 27, 2023

Ubuntu Commands

List files/directories and change path

For windows: install mobaxterm

1. $ `pwd` → Print Working Directory
2. $ `ls` → List : print the list of files and directories in current path
3. $ `ls <targetDirPath>` → List : print the list of files and directories in the targeted directory Path
4. $ `cd` → Change working directory to Home directory.
5. $ `cd` <targetDirPath> → Change working directory to targeted directory
6. $ `cd .` → Change to **Current** directory
7. $ `cd ..` → Change to **Parent** directory

Make/Delete/Copy a file/directory

- $ `cp <srcFilePath> <destFilePath>` → COPY a *File* at srcFilePath to destFilePath
- $ `cp -r <srcDirPath> <destDirPath>` → COPY a directory
- $ `exit, ^d` → EXIT an ongoing program
- $ `mkdir <directoryName>` → MAKE the directory
- $ `rmdir <directoryName>` → REMOVE the directory
- $ `rm <fileName>` → REMOVE the file fileName
- $ `rmdir <directoryName>` → REMOVE the directory
- $ `rm -r <directoryName>` → REMOVE the directory
- $ `mv <srcFilePath> <destFilePath>` → MOVE the file

Printing Contents of a File

1. $ `cat <fileName>` → whole content
2. $ `head <fileName>` → HEAD of the file
3. $ `man <cmdName>` → show MANUAL of cmdName
4. Press "`q`" to Quit
5. $ `top` → Display ongoing programs
6. $ `kill -9 <programID>` → Kill the program with id programID
7. others– $ `wget, time`,

Basic input/output from/to a file

```c
// Program to compute average of two float variables
#include<stdio.h>

float average(float a, float b){
    return ((a+b)/2.0);
}

int main(){
    float a, b, avg;

    scanf("%f %f", &a, &b);  // taking input from terminal
    avg = average(a, b);    //Compauting avarage
    printf("%f",avg); //writing on terminal
    return 0;
}
```

```c
// Program to compute average of two float variables
#include<stdio.h>

float average(float a, float b){
    return ((a+b)/2.0);
}

int main(){
    float a, b, avg;

    scanf("%f %f", &a, &b);  // taking input from terminal
    avg = average(a, b);    //Compauting avarage
    printf("%f",avg); //writing on terminal
    return 0;
}
```

- Sometimes input is large–
- Sometime we have many inputs
- embedding data directly into the source code– **a bad idea and Not practical**
- We require to take input data from files.

fscanf and fprintf

- fscanf and fprintf works same as scanf and printf

```
// Program to learn basic file operation
#include<stdio.h>

float average(float a, float b){
    return ((a+b)/2.0);
}

int main(){
    float a, b, avg;

    FILE * inp_file_ptr, * out_file_ptr; //File type pointer must be declared

    inp_file_ptr = fopen("input_file.txt","r"); // Opening input file for
        reading
    fscanf(inp_file_ptr, "%f %f", &a, &b);  // taking input from file
    fclose(inp_file_ptr);   // closing the input file

    avg = average(a, b);    //Compauting avarage

    out_file_ptr = fopen("output_file.txt","w");
    fprintf(out_file_ptr, "%f",avg); //writing on output file
    fclose(out_file_ptr); //closing the output file

    return 0;
}
```

Command Line Arguments

Why inputs from command line

- Another form of input
- Useful when you want to control your program from outside.
- To override defaults and have more direct control over the application

Example:

```
int main(int argc, char *argv[]) {
  /* ... */
}
```

or

```
int main(int argc, char **argv) {
  /* ... */
}
```

```c
// Program to compute average of two float variables
#include<stdio.h>
#include<stdlib.h> //that contains atof

float average(float a, float b){
    return ((a+b)/2.0);
}
int main(int argc, char *argv[]){
    float a, b, avg;
    if (argc==3){
        a = atof(argv[1]); //converting string to float
        b = atof(argv[2]);
    }else{
        scanf("%f %f", &a, &b);  // taking input from terminal
    }
    avg = average(a, b);    //Compauting avarage
    printf("%.2f",avg); //writing on terminal
    return 0;
}
```

```c
// Program to compute average of two float variables
#include<stdio.h>
#include<stdlib.h> //that contains atof

float average(float a, float b){
    return ((a+b)/2.0);
}
int main(int argc, char *argv[]){
    float a, b, avg;
    if (argc==3){
        a = atof(argv[1]); //converting string to float
        b = atof(argv[2]);
    }else{
        scanf("%f %f", &a, &b);  // taking input from terminal
    }
    avg = average(a, b);    //Compauting avarage
    printf("%.2f",avg); //writing on terminal
    return 0;
}
```

- argc (ARGument Counter): is The number of command-line arguments passed. It includes the name of the program
- argv (ARGument Vector): An array of strings pointers listing all the arguments.
- argv[0] is the name of the program , After that till argv[argc-1] every element is command-line arguments.
- Only strings can be taken from command line.

Compiling C program