Basic C/C++ Programming under GNU Linux and Microsoft Windows – Part 2

Praseed Pai

praseedp@gmail.com

The Character Count Program (sixth.cpp)

```
// sixth.cpp
// Program to Count # of characters
// One is expected to use I/O redirection operator
// ( < ) to give the input file name.
// g++ -o file_stat.exe sixth.cpp
// ./file stat.exe < sixth.cpp
// cl /Fefile stat.exe sixth.cpp
// file stat.exe < sixth.cpp
#include <stdio.h>
int main( int argc , char **argv ){
  int ccount=0; // character count
  int c = 0; // next character
  // while not end of the file ( use I/O redirection for the input )
  while (( c = getchar() ) != EOF ){ ccount++; }
  printf("Character Count = %d\n",ccount);
```

The Character Count/Line Count Program (seventh.cpp)

```
// seventh.cpp
// Program to Count # of characters and Lines
// One is expected to use I/O redirection operator
// ( < ) to give the input file name.
// g++ -o file stat.exe seventh.cpp
// ./file stat.exe < seventh.cpp
// cl /Fefile stat.exe seventh.cpp
// file stat.exe < seventh.cpp
#include <stdio.h>
int main( int argc , char **argv ){
  int ccount=0; // character count
  int Incount = 0;
  int c = 0; // next character
  // while not end of the file ( use I/O redirection for the input )
  while (( c = getchar() ) != EOF ){
     ccount++;
           if ( c == '\n' ) { Incount++; }
  printf("Character Count = %d\n",ccount);
  printf("Line count = %d\n", Incount );
```

The Character/Word/Line Count (eight.cpp)

```
// eight.cpp
// Program to Count # of characters , words and Lines
// One is expected to use I/O redirection operator
// ( < ) to give the input file name.
// g++ -o file_stat.exe eight.cpp
// ./file stat.exe < eight.cpp
// cl /Fefile stat.exe eight.cpp
// file stat.exe < eight.cpp
// Adapted from K&R book (2nd Edition Page #20)
         Section 1.5.4 - Word counting
#include <stdio.h>
#define IN 1
#define OUT 0
int main( int argc , char **argv ){
  int ccount=0; // character count
  int Incount = 0; // line count
  int wccount = 0; //word count
  int c = 0; // next character
  int state = OUT;
```

```
// while not end of the file ( use I/O redirection for the input )
while (( c = getchar() ) != EOF ){
    ccount++;
        if ( c == '\n' ) { Incount++; }
        if ( c == '\' || c == '\t' || c == '\n' ) {
        state = OUT;
        } else if ( state == OUT ) { state = IN; wccount++; }
}
printf("Character Count = %d\n", ccount);
printf("Line count = %d\n", Incount );
printf("Word count = %d\n", wccount );
```

Topology of the Command line

```
E:\TCPP>third
argv[0]--> [t][h][i][r][d][\0]
E:\TCPP>third.exe
argv[0]--> [t][h][i][r][d][.][e][x][e][\0]
F.\TCPP>third one 2 three
argv[0]--> [t][h][i][r][d][\0]
argv[1]--> [o][n][e][\0]
argv[2]--> [2][\0]
argv[3]--> [t][h][r][e][e][\0]
E:\TCPP>third 1 2 3
argv[0]--> [t][h][i][r][d][\0]
argv[1]--> [1][\0]
argv[2]--> [2][\0]
argv[3]--> [3][\0]
```

```
E:\TCPP>third 1 24567 3
argv[0]--> [t][h][i][r][d][\0]
argv[1]--> [1][\0]
argv[2]--> [2][4][5][6][7][\0]
argv[3]--> [3][\0]

E:\TCPP>third 1 24567 3 helloworld
argv[0]--> [t][h][i][r][d][\0]
argv[1]--> [1][\0]
argv[2]--> [2][4][5][6][7][\0]
argv[3]--> [3][\0]
argv[4]--> [h][e][l][l][o][w][o][r][l][d][\0]
```

Spit the Topology of the Command line arguments (nine.cpp)

```
//nine.cpp
//A simple program which spits the
//command line arguments in a form
//which C/C++ startup code stores
//command line argument in an array.
//This program is to be run with
//various command line arguments.
// g++ -o nine.exe nine.cpp
// cl /Fenine.exe nine.cpp
#include <stdio.h>
int main( int argc , char **argv ){
 for(int i=0;i<argc;++i ){
   char *temp = argv[i];
   printf("argv[%d]--> ",i);
   char *itemp = temp;
   while (*itemp != 0){
      printf("[%c]",*itemp);
      itemp++;
   } // end while
   printf("[\\0]\n");
 } // end for
```

Spit the Environment Variables (ten.cpp)

```
// ten.cpp
// A C/C++ program to spit the environmental variables
// g++ -oenvspitter.exe ten.cpp
// cl /Feenvspitter.exe ten.cpp
#include <stdio.h>
int main( int argc , char **argv , char **envp ){
 char **temp = envp;
 while (*(temp+1) != 0 ) {
       puts(*temp);
      temp++;
 return 0;
```

Questions

- The Source code is available @ https://github.com/praseedpai/BasicCppCourse/tree/main/Part2
- The Slide is also available as PDF from the Above URL
- Any?