C Programming Basics

SDS 322/392

November 24, 2015

Email any questions to:
rauta@tacc.utexas.edu

THE UNIVERSITY OF TEXAS AT AUSTIN

Overview of the Lecture

- Writing a Basic C Program
- Understanding Errors
- Comments, Keywords, Identifiers, Variables
- Standard Input and Output
- Operators
- Control Structures
- Functions in C
- Arrays, Structures
- Pointers
- Working with Files
- Misc. Topics

All the concepts are accompanied by examples.

File I/O

- File pointer is required for accessing files to read, write or append

```
FILE *fp;
```

- **fopen** function is used to open a file and it returns a file pointer

```
FILE *fopen(const char *filename, const char *mode);
```

- The modes in which a file can be opened

```
r  - open for reading
w  - open for writing (file need not exist)
a  - open for appending (file need not exist)
r+ - open for reading and writing, start at beginning
w+ - open for reading and writing (overwrite file)
a+ - open for reading and writing (append if file exists)
```

- To close a file

```
int fclose(FILE *a_file);
```

Review the links: http://www.cprogramming.com/tutorial/cfileio.html
http://www.codingunit.com/c-tutorial-binary-file-io

File I/O: fileExample.c

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
  int i, myInt;
  FILE *ifp;
  char *mode = "r";
  ifp = fopen("in.txt", mode);
  if (ifp == NULL) {
    fprintf(stderr, "Can't open input file in.txt!\n");
    exit(1);
  }else{
    for (i=0; i<10; i++){
      fscanf(ifp,"%d", &myInt);
      printf("%d\n",myInt);
    }
  }
  fclose(ifp);
  return 0;
}
```

<--- `fscanf` is used for reading file contents

What if you do not know the number of lines in a file?

Hint: Use another type of loop

Write to a File: writeToFile.c

```c
#include <stdio.h>
int main(){
    FILE *fp;
    fp = fopen("in2.txt","a+");
    fprintf(fp,"\n%d",7000);
    fclose(fp);
    return 0;
}
```

Opening the file in append mode

fprintf is used for writing data to a file

User-Defined Header Files

- Useful in multi-module, multi-person software development effort
- Save the following code in a file named `head.h` and don't compile/run it

```
/* This is my little header file named head.h */
#define HAPPY 100
#define SPIT printf
#define POOL {
#define PEEL }
```

User-Defined Header Files

- This is how the file `head.h` can be included in any program, here `headTest.c`

```
#include <stdio.h>
#include "head.h" <-- Notice the quotes around file name
int main()
POOL
SPIT("This guy is happy: %d percent\n", HAPPY);
return(0);
PEEL
```

```
Output:
This guy is happy: 100 percent
```

Miscellaneous Topic

- We are mostly done with the C part of the course
  - Will cover math libraries briefly now
  - Will revisit the topic of math libraries at a later point if schedule permits

Including Library File for Maths: mathExample.c

```
#include <stdio.h>
#include <math.h>
int main(){
 double myNum = 2.2;
 int times = 8 ;
 printf("Square root of %lf is: %lf\n",myNum, sqrt(myNum));
 return 0;
}
```

```
Output:
Square root of 2.200000 is: 1.483240
```

Math Library on Stampede (MinGW users can ignore this)

- Note that `math.h` is a header-file that just includes the declarations of the math functions – recall function prototypes
- The compiled definitions are in the math library
- Link your program to the math library by adding `-lm` to the `gcc` command, no need for doing this if you are using `icc`

```
login3$ gcc -o mathExample mathExample.c
/tmp/ccOdwtYH.o: In function `main':
mathExample.c:(.text+0x2b): undefined reference to `sqrt'
mathExample.c:(.text+0x74): undefined reference to `pow'
collect2: ld returned 1 exit status
login3$ gcc -o mathExample mathExample.c -lm
login3$
```

In-Class Exercise: File Copy

- Write a program to copy the contents of one file to another
  - The name of the input file is **`aFile.txt`**
  - The name of the output file is **`bFile.txt`**
  - Contents of **`aFile.txt`** are:

This is a fun exercise!

I am learning file I/O.

I have to finish the homework soon.

Solution to File Copy - 1

```
#include <stdio.h>
int main(){
 FILE *myfile, *outfile;
 char line[200];
 myfile = fopen("aFile.txt", "r");
 outfile = fopen("bFile.txt", "w");
 if (myfile != NULL){
   while (fgets(line, sizeof(line), myfile )!= NULL ) {
     fprintf(outfile,line);
     printf("%s\n", line);
   }
 }else{
     printf("\nUnable to open file\n");
 }
 return 0;
}
```

Solution to File Copy - 2

```
#include <stdio.h>
int main(){
        FILE *myfile, *outfile;
        int c;
        myfile = fopen("aFile.txt", "r");
        outfile = fopen("bFile.txt", "w");
        if (myfile != NULL){
           c = getc(myfile) ;
           while ( c != EOF)  {
                 putc(c,  outfile);
                 c =  getc( myfile ) ;
           }
           fclose(myfile);
           fclose(outfile);

        } else{
           printf("\nUnable to open file\n");
        }
        return 0;
}
```

Redoing Matrix Multiplication

- Redo the matrix multiplication problem assigned in the previous exercises using file I/O

- Instead of hard-coding the initial values of matrices, read them from a file

- Instead of printing the product of the two matrices on the screen, print it to the file

References

- Pointers in C, Yashavant Kanetkar
- http://www.eskimo.com/~scs/cclass/int/sx9b.html
- http://stackoverflow.com/questions/1169858/global-memory-management-in-c-in-stack-or-heap