IO in C

Discussion

- Why do we need input output?

Input/Output

- Console I/O

1. Formatted
2. Unformatted

- File I/O

Formatted Console I/O

- scanf and printf

Syntax:

printf(“format string”, list of variables)

scanf(“format string”, list of addresses of variables)

Format string can contain:

- Characters which can be printed as they are (for printf)
- Conversion specifications that begin with a % sign
- Escape sequences that begin with / sign

Format specifications

- Integer: %d, %u, %x, %o, %ld ,%lu
- Real: %f, %lf
- Character: %c
- String: %s

Escape sequences

- \n - new line
- \b - backspace
- \t - tabsapce
- \r - carriage return
- \a - alert
- \’ - single quote
- \” - double quote
- \\ - backward slash

sprintf and sscanf

- sprintf: stores output or prints into an array of characters
- sscanf: takes input from a string (array of characters) and stores it into variables as per the format string

Unformatted Console I/O

- getch()
- getche()
- getchar()
- fgetchar()
- putch()
- putchar()
- fputchar()

gets and puts

- ‘gets’ gets a newline terminated string of characters and replaces \n with \0
- ‘puts’ prints a whole string on the screen

Disadvantage:

input and output only one string at time unlike scanf and printf

Reading from/Writing to file

- File pointer FILE *fp
- fopen()
- fgetc()/fputc()
- EOF
- fclose()

File opening modes

- r-reading from file
- w-writing to file
- a-appending to file

argc and argv

- argc: It’s an int value that indicates the argument count passed at the command line
- argv: It’s an array of pointers to strings passed on the command line

Eg:

filecopy file1.c file2.c

argc: 3

argv[0] = filecopy

argv[1]= file1.c argv[2]=file2.c

Standard IO Devices

- Three standard files

1. stdin
2. stdout
3. stderr

IO Redirection: <,>

Summary

- We can pass parameters to a program at command line using the concept of ‘command line arguments’.
- The command line argument **argv** contains values passed to the program, whereas, **argc** contains number of arguments.
- We can use the standard file pointer **stdin** to take input from standard input device such as keyboard.
- We can use the standard file pointer **stdout** to send output to the standard output device such as a monitor.
- We can use the standard file pointers **stdprn** and **stdaux** to interact with printer and auxiliary devices respectively.
- Redirection allows a program to read from or write to files at command prompt.
- The operators **<** and **>** are called redirection operators.

Summary (cont..)

- All I/O in C is done using standard library functions.
- There are several functions available for performing console input/output.
- The formatted console I/O functions can force the user to receive the input in a fixed format and display the output in a fixed format.
- There are several format specifiers and escape sequences available to format input and output.
- Unformatted console I/O functions work faster since they do not have the overheads of formatting the input or output.

Questions

- What is the optput if value 25 is passed on to scanf

printf(“%d”,scanf(“%d”,&i));

- How will you use the following program to copy contents of one file to another

```
main()
{ char ch, str[10];
while ((ch=getc(stdin))!=1)
putc(ch,stdout);}
```

- What do c and v in argc and argv stand for
- myprog one two three

```
main(int argc, char *argv[])
{
        int i;
        for(i=1;i<=3;i++)
        printf("%c",*argv[i]);
}
```

Answers

- 1
- filename<sourcefile>targetfile
- count of arguments and vector(array) of arguments
- ott

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com