I/O in C

Hyung-Sin Kim

Computing Foundations for Data Science

Graduate School of Data Science, Seoul National University

One Character I/O

- putchar – print a single character (an input is assumed to be an ASCII value)
  - char c = ‘h’;
  - putchar(c);
  - putchar(‘h’);
  - putchar(104);

- getchar – get a single character input (returns its ASCII value)
  - char c;
  - c = getchar();

Buffered I/O

- Reading each character right after a keyboard button is pushed is error-prone
    - To address the problem, keyboard input is buffered until we press Enter
    - Before Enter is pressed, you can modify the input stream freely
    - When Enter is pressed, the whole buffered input including ‘\n’ is delivered to the program

- Output is buffered too.
    - Output from putchar is buffered until it prints out ‘\n’

Buffered I/O

- Example 1

```
#include <stdio.h>

int main(void) {
    char c1;
    char c2;
    printf("Input char1:\n");
    c1 = getchar();
    printf("Input char2:\n");
    c2 = getchar();
    printf("char1 is %c and char2 is %c\n", char1, char2);
    return 0;
}
```

- Example 2

```
#include <stdio.h>
#include <unistd.h>

int main(void) {
    putchar('s');

    sleep(5);

    putchar('s');
    putchar('\n');
    return 0;
}
```

Formatted I/O – printf

- **printf** prints out ASCII text embedded with values
    - In doing so, it must convert any non-ASCII value, such as integer, into an ASCII value
- printf(”format string”, values);
    - Format string consists of normal characters, special characters, and conversion specifications
    - printf examines each character in the format string sequentially
        - If the character is a normal character, it simply print this out
        - If the character is ‘%’, it recognizes a conversion specification, such as %d. Then, the next character indicates how the next pending parameter should be interpreted
        - If the character is ‘\’, it recognizes a special character, such as ‘\n’

Formatted I/O – printf

- printf conversion specifications

printf Conversions	Printed as
%d, %i	Signed decimal
%o	octal
%x, %X	Hexadecimal (a-f or A-F)
%u	Unsigned decimal
%c	Single character
%s	String, terminated by \0
%f	Floating point in decimal notation
%e, %E	Floating point in exponential notation
%p	Pointer

Formatted I/O – scanf

- scanf reads formatted ASCII data and **converts** it into another data type, if needed
  - In doing so, it must convert any non-ASCII value, such as integer, into an ASCII value
- scanf(”format string”, &variable or pointer);
- Example
  - int main {
  -     char name[50];
  -     int bd_year;
  -     int bd_month;
  -     int bd_day;
  -     printf(“Enter : last name and birthday YYYY/MM/DD\n”);
  -     scanf(“%s: %d/%d/%d, name, &bd_year, &bd_month, &bd_day);
  -     printf(“Name: %s\n”,);
  -     printf(“Birthday: %d/%d/%d, bd_year, bd_month, bd_day);
  - }

Formatted I/O – scanf

- scanf conversion specifications

scanf Conversions	Parameter type
%d	Signed decimal
%i	Decimal, octal (leading 0), hex (leading 0x or 0X)
%o	octal
%x	Hexadecimal
%u	Unsigned decimal
%c	Single character
%s	String of non-white space characters, \0 added automatically
%f, %e	Floating point number
%lf	Double precision floating point number

I/O from Files

- File pointer – a pointer that points to a type FILE
  - FILE *filePtr;

- Opening a file
  - filePtr = fopen(“file name”, “mode”)
  - fopen() returns a file pointer to the physical file “file name”
  - Modes
    - r: reading
    - w: writing
    - a: appending
    - r+: reading and writing
  - Good practice: checking if the fopen call was successful
    - if (filePtr == NULL)
    -     printf(“fopen error!\n”);

I/O from Files

- fscanf(filePtr, “format string”, variables);
  - Reading a file (filePtr)
- fprintf(filePtr, “format string”, variables);
  - Writing to a file (filePtr)
- Example
  - FILE *infile;
  - FILE *outfile;
  - char str[50];
  -
  - infile = fopen(“input.txt”, “r”);
  - outfile = fopen(“output.txt”, “w”);
  - while (fscanf(infile, “%s”, str) != EOF)
  -     fprintf(outfile, “%s ”, str);

Questions?

Thanks!