Computing Bootcamp

File I/O in C

Lecture 32

Hyung-Sin Kim

SNU Graduate School of Data Science

I/O from Files

- File pointer – a pointer that points to a type FILE
  - FILE *filePtr;
  - It moves forward as you read or write

- 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

- char a = fgetc(filePtr);
  - Read and return a single character from a file, where **filePtr** points
  - Return a special value **EOF** when no more data can be read
- fputc(a, filePtr);
  - Write a single character **a** to a file, where **filePtr** points

- fscanf(filePtr, “format string”, variables);
  - Read a file, where **filePtr** points
  - Store the read values to the corresponding variables
  - Return a special value **EOF** when no more data can be read
- fprintf(filePtr, “format string”, variables);
  - Writing the variables’ values to a file, where **filePtr** points

I/O from Files – Example

```
#include <stdio.h>

int main(void) {
    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);
}
```

Summary

- Buffered Character I/O
  - putchar
  - getchar
- Formatted I/O
  - printf
  - scanf
- I/O from Files
  - fgetc
  - fputc
  - fscanf
  - fprintf

Thanks!