

창의적 소프트웨어 설계



11주차 실습 – File IO

노인우, inwoo13@hanyang.ac.kr

한중수, soohan@hanyang.ac.kr

Overview

- ◆ C stdio File Interface
- ◆ C++ File Stream Interface
- ◆ Closing Files

C stdio File Interface

◆ fopen, fread, fclose

```
#include <stdio.h>

int main() {
    FILE* fp;
    errno_t et = fopen_s(&fp, "test.txt", "r");
    if (fp == NULL) return -1; // Error in opening the file.

    char buf[256];
    size_t read = fread(buf, 256, 10, fp); // sizeof(buf)
    printf("%s", buf);

    fclose(fp);

    return 0;
}
```

C stdio File Interface

```
#include <stdio.h>

int main() {
    FILE* fp_read = fopen("source.txt", "r");
    FILE* fp_write = fopen("destination.txt", "w");

    if (fp_read == NULL || fp_write == NULL) {
        return -1;
    }

    char buf[1024] = "";
    size_t read = 0;

    while ((read = fread(buf, 1, 1024, fp_read)) > 0) {
        size_t written = fwrite(buf, read, 1, fp_write);
    }

    printf("%s", buf);

    fclose(fp_read);
    fclose(fp_write);

    return 0;
}
```

C stdio File Interface

```
#include <stdio.h>

int main() {
    FILE* fp_read = fopen("source.txt", "r");
    FILE* fp_write = fopen("destination.txt", "w");

    if (fp_read == NULL || fp_write == NULL) return -1;

    int data;

    while (fscanf(fp_read, "%d", &data) > 0) { // buffer can only contain numerical
        fprintf(fp_write, "%d\n", data);
    }

    fclose(fp_read);
    fclose(fp_write);

    return 0;
}
```

C stdio File Interface

- ◆ 키보드로 입력 받은 값을 file에 출력하는 메모장
 - 입력 함수는 무엇을 사용해도 무관
 - return 키로 줄바꿈
 - wq + return 입력으로 종료
- ◆ 체크포인트:
 - 한 번에 입력 가능한 크기에 한계가 있는가?
 - 전체 입력 크기에 한계가 있는가?

C++ File Stream Interface

◆ fstream

```
#include <fstream>

using namespace std;

int main() {
    ifstream inStream;
    ofstream outStream;

    inStream.open("source.txt");
    outStream.open("destination.txt");

    char buf[256] = "";
    inStream >> buf;
    outStream << buf;

    return 0;
}
```

Checking File Open Success

- ◆ File과 같은 코드 외부의 데이터의 결과는 매번 다를 수 있음
- ◆ 오류 허용(Fault Tolerant)한 코드 작성 필요

```
#include <iostream>
#include <fstream>

using namespace std;

int main() {
    ifstream inStream;
    ofstream outStream;

    inStream.open("source1.txt");

    if (inStream.fail()) {
        cout << "Input file open failed" << endl;
        exit(1);
    }

    outStream.open("destination.txt");
    outStream.close();

    return 0;
}
```


참고자료

1. class vs typename,

<https://stackoverflow.com/questions/213121/use-class-or-typename-for-template-parameters>

2. typename, Stan Lippman,

<https://web.archive.org/web/20060619131004/http://blogs.msdn.com/slippman/archive/2004/08/11/212768.aspx>



File Permission

◆ DAC (Discretionary Access Control)

■ https://en.wikipedia.org/wiki/Discretionary_access_control