Assignment1

저수준과 고수준 파일 입출력의 차이

사이버보안학과

201620641

유 상 정

1. **저수준 파일 입출력이란?**

유닉스에서 파일을 읽고 쓰는 방법은 저수준 파일 입출력과 고수준 파일 입출력으로 구분할 수 있다. 우선, 저수준 파일 입출력에 대해 말해보겠다.

저수준 파일 입출력은 유닉스 커널의 시스템을 호출하여서 파일 입출력을 수행한다. 그래서 파일에 좀 더 빠르게 접근할 수 있다는 장점이 있다. 또한 파일을 byte 단위로 읽고 써서 일반 파일 뿐만 아니라 특수 파일도 읽고 쓸 수 있다. 하지만 byte 단위로만 파일 입출력을 수행하므로, 이를 이용해서 프로그램을 작성하려면 byte를 적당한 형태의 데이터로 캐스팅하는 함수와 같은 여러 추가적인 기능을 구현해야 한다.

1. **고수준 파일 입출력이란?**

저수준 파일 입출력이 byte를 적당한 형태의 데이터로 캐스팅해야 하는 단점을 해결하기 위해서 제공되는 것이 고수준 파일 업출력이다. 고수준 파일 입출력은 C 언어의 표준 함수로 제공되고, 데이터를 byte 단위로 한정하지 않고 buffer를 이용해서 한꺼번에 읽기와 쓰기를 수행한다. 또한 다양한 입출력 data 변환 기능도 이미 구현되어 있어서 data형에 따라 편리하게 이용할 수 있다. 또한 고수준 파일 입출력을 표준 입출력 라이브러리라고도 한다.

1. **저수준 파일 입출력과 고수준 파일 입출력의 차이점 (이론적인 측면)**

저수준 파일 입출력은 파일을 참조할 때, fd(file descriptor)를 사용하고, 고수준 파일 입출력은 fp(file pointer)를 사용한다.

(참고: fp는 C언어에서 제정한 표준 입출력 객체, fd는 OS에서 제공하는 입출력 식별자)

저수준 파일 입출력이 훨씬 빠르고, byte단위로 읽고 써서 특수 파일에 접근이 가능하다. 또한 세밀한 조작이 고수준보다 더 쉽다.

고수준 파일 입출력은 사용하기 쉽고, buffer 단위로 읽고 쓴다(stream을 입출력). Data의 입출력 동기화가 쉽다. 또한 여러가지 형식을 지원한다. (즉, 기능이 더 많다.)

주요 함수 차이

저수준 파일 입출력

* open, close, read, write, dup, dup2, fcntl, lseek, fsync

고수준 파일 입출력

* fopen, fclose, fread, fwrite, fputs, fgets, fprintf, fscanf, freopen, fseek

차이점으로는 위와 같이 기술한 것들이 있다. 다만 위에서 저수준의 입출력이 훨씬 빠르다고 하였는데, 실제로 파일 입출력을 한다면, 저수준이 더 효율이 안 좋을 수도 있다. 왜냐하면, 저수준은 byte단위로 파일을 입출력 하는데, 만약 file이 100000byte정도 된다면, 저수준 파일 입출력은 100000번 수행해야 한다. 하지만, 고수준은 buffer크기만큼 한 번에 수행하기 때문에, 효율이 더 좋을 수도 있다. 하지만 실제 속도의 차이는 직접 테스트해보지 않으면, 정확히 알 수 없다.

1. **저수준 파일 입출력과 고수준 파일 입출력의 차이점 (실제 측면)**

실제로 코딩을 해본 결과, 저수준 파일 입출력은 고수준 파일 입출력에 비해서 매우 불편하다. 왜냐하면, 고수준 파일 입출력은 많은 함수를 제공한다. 특히, 이번 과제에서 fscanf의 장점에 반했다. scanf처럼 자기가 원하는 타입으로 data의 내용을 받을 수 있다. 하지만, 저수준 파일 입출력은 자신이 원하는 대로 받지 못하고, 모두 char형으로 받아야한다. 그래서 따로 parsing을 해야 하는 단점이 명백히 드러났다. 이 방법에는 두가지가 존재한다. (물론 더 있을 것이다.) 첫째, 본인이 쓴 방법으로서, buf로 한 라인을 받아서 strtok함수를 사용한다. 그래서 각 문자를 “ “ 기준으로 자르고, 학번을 제외한 것들을 atoi()함수를 사용해서 int형으로 변환해준다. 두번째 방법으로는 read함수로 받을 때, 학번이면 학번, 각 점수별로 원하는 만큼만 read를 하는 방식이다. 명백한 것은 위 두 방법 모두, 고수준 파일 입출력보다 불편하다는 것이다. 또한 헤더파일에서도 차이가 존재했다. 고수준 파일 입출력은 header <stdio.h>에서 기능을 모두 제공해준다. 하지만, 저수준 파일 입출력은 open을 하기위해서 header <fcntl.h>를 선언하고, close를 하기위해서 <unistd.h>를 선언 해야 한다. 이와 같이, 실제 코딩을 하는 입장에서 고수준 파일 입출력이 훨씬 쓰기 편함을 알 수 있다.

1. **각 소스코드로 컴파일한 실행화면**

* prog1 (저수준 파일 입출력)



* Prog2 (고수준 파일 입출력)



각 소스코드의 실행결과는 차이가 없다.

* 참고 사이트

[시스템프로그래밍] 저수준 파일 입출력(<https://12bme.tistory.com/211>)

파일 포인터 와 파일 디스크립터 차이

(<https://atomkwang.tistory.com/entry/%ED%8C%8C%EC%9D%BC-%ED%8F%AC%EC%9D%B8%ED%84%B0%EC%99%80-%ED%8C%8C%EC%9D%BC-%EB%94%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%84%B0-%EC%B0%A8%EC%9D%B4%EC%A0%90>)