

Lab 12

2024학년도 2학기 리눅스시스템

조교 이지원

lee.jiwon@sookmyung.ac.kr



SOOKMYUNG WOMEN'S UNIVERSITY



Lab 12. 파일 시스템과 파일 입출력 실습

1. 문서 편집

- 1) test.txt 파일을 만들어 이 파일에 대해 `$ls -sl` 명령을 사용하고 그 출력 결과를 설명한다.
- 2) test.txt 파일에 대해 `stat` 명령어를 사용하고 그 출력 결과를 설명한다.
- 3) test.txt 파일에 대해 `$ls -li` 명령을 사용하고 i-노드 번호를 확인한다.
- 4) test.txt 파일에 대해 `touch` 명령어를 사용하고 사용 전 후의 파일 속성을 비교하여 설명한다.

2. 링크

- 1) test.txt 파일에 대한 하드 링크와 심볼릭 링크를 만든다.
- 2) 만들어진 하드 링크와 심볼릭 링크의 차이점을 설명한다.
- 3) `$ls -i` 명령을 이용하여 각각의 i-노드 번호를 확인한다.

3. cat 명령어 확장 (cat.c)

- fgets() 함수를 사용하지 않고 교재의 프로그램 12.1을 확장하여 구현하시오.

- cat 명령어 확장 구현

\$ cat 파일명*

- 명령줄 인수로 여러 개의 파일을 받으면 순차적으로 파일 내용을 출력한다.
- 명령줄 인수가 없을 경우, 표준입력으로부터 입력받은 내용을 표준출력에 출력한다

- cat -n 옵션 구현

- 파일 내용을 줄 번호와 함께 출력한다.

\$ cat -n [파일명]

- 입력 파일 개수는 하나로 제한한다.
- 명령줄 인수가 없을 경우, 표준입력으로부터 입력받은 내용을 표준출력에 출력한다
- 실제 명령의 동작과 비교해볼 것.

3. 실행화면 참고 (\$ cat 명령어)

- 파일 1개 입력

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./cat file.txt
Linux System!
This is Lab12.
```

- 파일 2개 입력

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./cat file.txt file2.txt
Linux System!
This is Lab12.
Nice to meet you!
My name is Jiwon.
```

- 파일이 제대로 열렸는지 검사 (file1.txt 존재하지 않음.)

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./cat file1.txt
File Error: file1.txt
```

- 명령줄 인수가 없을 경우, 표준입력으로부터 입력받은 내용을 표준출력에 출력한다. (Ctrl+D 종료)

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./cat
hi
  hi
hello
  hello
goodbye!
  goodbye!
```

3. 실행화면 참고 (\$ cat -n 명령)

- 파일의 줄 수 출력

```
u23000000@ubuntu-virtualbox:~/linux/chap12$ ./cat -n file.txt
  1 Linux System!
  2 This is Lab12.
```

- 파일이 제대로 열렸는지 검사 (file1.txt 존재하지 않음.)

```
u23000000@ubuntu-virtualbox:~/linux/chap12$ ./cat -n file1.txt
File Error: file1.txt
```

- n이 아닌 다른 옵션을 준 경우 안내

```
u23000000@ubuntu-virtualbox:~/linux/chap12$ ./cat -o
Error: Invalid option '-o'. Please use the -n option.
u23000000@ubuntu-virtualbox:~/linux/chap12$ ./cat -o file.txt
Error: Invalid option '-o'. Please use the -n option.
```

- 명령줄 인수가 없을 경우, 표준입력으로부터 입력받은 내용을 표준출력에 출력한다. (Ctrl+D 종료)

```
u23000000@ubuntu-virtualbox:~/linux/chap12$ ./cat -n
hi
  1 hi
hello
  2 hello
goodbye!
  3 goodbye!
```

3. 교재 cat.c

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[]) {
4     FILE *fp;
5     int c;
6
7     if(argc < 2)
8         fp = stdin;    //명령줄 인수가 없으면 표준입력 사용
9     else
10        fp = fopen(argv[1], "r"); //읽기 전용으로 파일 읽기
11
12    c = fgetc(fp);      //파일로부터 문자 읽기
13    while(c != EOF) {   //파일 끝이 아니면
14        fputc(c, stdout); //읽은 문자를 표준출력에 출력
15        c = fgetc(fp);  //파일로부터 문자 읽기
16    }
17    fclose(fp);
18    return 0;
19
20 }
```


4. wc 명령어 구현 (wc.c)

- cat 명령어를 확장하여 줄 수, 단어 수, 문자 수를 출력하는 wc 명령어를 구현하시오.

\$ wc [파일명] 의 동작과 똑같이 만들면 됨.

// 입력 파일 개수는 하나로 제한한다.

// 줄 수, 단어 수, 문자 수 사이에 tab으로 공백을 둔다

실제 wc의 동작과 비교해볼 것.

- 줄 수는 \n의 개수를 센다.
- 빈칸과 빈칸 사이의 단어 개수를 센다.
- 읽은 문자 개수를 센다.

4. 실행화면 참고

- 파일의 줄 수, 단어 수, 문자 수 출력

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./wc file.txt
  2      5     29    file.txt
```

- 파일이 제대로 열렸는지 검사 (file1.txt 존재하지 않음.)

```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./wc file1.txt
File Error: file1.txt
```

- 명령줄 인수가 없을 경우, 표준입력으로부터 입력받은 내용을 표준출력에 출력한다. (Ctrl+D 종료)


```
u2300000@ubuntu-virtualbox:~/linux/chap12$ ./wc
hi
hello
goodbye!
  3      3     18
```



Lab 12. 과제 설명

과제 제출

■ 과제 제출 양식1 (보고서)

1. 자신의 학번으로 된 계정으로 로그인하여 실습을 진행한 후 캡처 
2. 각 문제별 실행 화면 캡처 및 설명을 보고서로 작성
3. 보고서에 실습번호, 분반, 학과, 학번, 이름을 적을 것 (보고서 기본 형식 유지)
4. 워드로 보고서를 작성한 뒤 PDF로 변환하여 제출
5. 파일명: 실습번호_분반_학번_이름 (Lab12_001_2300000_눈송이.pdf)

■ 과제 제출 양식2 (cat.c, wc.c)

1. 파일명 준수 (미준수 시 0점)
2. 각 파일에 주석으로 분반, 학과, 학번, 이름 반드시 적을 것

// 001 컴퓨터과학과 2300000 눈송이

■ 최종 과제 제출 양식 (Lab12_학번.zip)

1. pdf 파일(보고서)+cat.c+wc.c를 압축한 하나의 zip파일

과제 제출

- 과제 제출 방법
 - Snowboard Lab12에 pdf 파일(보고서) + cat.c + wc.c를 압축한 **하나의 zip파일** 제출
 - 제출 전 양식 준수 여부 반드시 확인!!!!!!
 - 제출 기한 : **1분반) 12월 3일 화요일 오후 11:59**
2분반) 12월 4일 수요일 오후 11:59
 - 제출 기한 이후 24시간 이내 추가 제출 가능 (20% 감점)

+) 과제 질문 방법

- lee.jiwon@sookmyung.ac.kr (조교 메일로 질문 보내기)
- 질문 답변 시간: 월-목 10:00-17:00
- 질문 시 주의사항

- 충분히 고민 후 질문 (질문하기 전 구글링 필수!)
- 메일에 반드시 과목, 분반, 전공, 학번, 이름 명시
- 몇 번 과제에서 어떤 부분이 막혔는지, 어떤 과정이 문제인지 **명확한 설명 첨부**
- 코드 질문 시 화면 캡처가 아닌 실제 코드 파일 첨부 (출력 결과는 캡처 가능)
- 답장이 늦을 수 있으니 여유 있게 미리 질문 (특히 과제 제출 마지막 날 유의!)