

[506489] 시스템프로그래밍

실습 #04 문제 및 보고서

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	01 (담당교수: 김태운)

<주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
 - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- **SmartLEAD 제출 데드라인:**
 - **월요일 분반: 다음 다음 실습 시간 전날까지(일요일 까지) // 2 주 과제**
 - **수요일 분반: 다음 다음 실습 시간 전날까지 (화요일 까지) // 2 주 과제**
 - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
 - 주말/휴일/학교행사 등 모든 날짜 카운트 함
 - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
 - 예외 없음
- **SmartLEAD 에 아래의 파일을 제출해 주세요**
 - **보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)**
 - 보고서 파일명에 이름과 학번을 입력해 주세요.
 - **소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)**

<개요>

이번 과제는 4 주차 강의 내용을 복습하는 내용입니다: 저수준 및 고수준 파일 입출력!

<실습 과제>

[Q 0] 요약 [배점: 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.
 답변: 이번 과제를 통해 저수준 파일 입출력과 고수준 파일 입출력을 비교 분석할 수 있어서
 좋았습니다. 또한, 파일 오프셋 관련 문제를 통해 '\n', '\0' 등 개행 문자에 대한 개념을 확실히
 잡을 수 있었습니다.

[Q 1] 저수준 파일 입출력 [배점: 30]

저수준 파일 입출력을 이용해 파일을 복사하는 프로그램을 작성하시오. '-c' 옵션을
 사용하면 파일을 암호화 할 수 있습니다.

- 프로그램명 : lowcopy
- 사용방법 1 : lowcopy 원본파일명 복사본파일명
 - 예: ./lowcopy original.txt new.txt 라고 입력하면 원본파일 original.txt
 파일과 동일한 사본을 생성하고, 사본의 이름은 new.txt 가 됩니다.
- 사용방법 2 : ./lowcopy -c 숫자 원본파일명 복사본파일명
 - 예: lowcopy -c 7 original.txt new.txt 라고 입력하면 original.txt
 파일에서 읽은 모든 바이트에 각각 7을 더한 뒤, 새로운 파일 new.txt 에 기록함.

[문제]

'a' 부터 'z' 까지 순차적으로 입력된 alphabet.txt 파일을 생성하세요.

[1] lowcopy alphabet.txt low_alpha.txt 입력하세요. 다음으로, cat
 low_alpha.txt 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[2] lowcopy -c 7 alphabet.txt crypt_alpha.txt 입력하세요. 다음으로, cat
 crypt_alpha.txt 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

소스코드도 제출해야 합니다.

답변 [1]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q01$ ./lowcopy alphabet.txt low_alpha.txt
yeongju@vm-ubuntu20:~/sp2021-2/w04/q01$ cat low_alpha.txt
abcdefghijklmnopqrstuvwxyz
```

답변 [2]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q01$ ./lowcopy -c 7 alphabet.txt crypt_alpha.txt
yeongju@vm-ubuntu20:~/sp2021-2/w04/q01$ cat crypt_alpha.txt
hijklmnopqrstuvwxyz{~}~♦♦
```

[Q 2] 고수준 파일 입출력 [배점: 30]

고수준 파일 입출력을 이용해 파일을 복사하는 프로그램을 작성하시오. ‘-d’ 옵션을 사용하면 ‘-c’ 옵션으로 암호화된 파일을 복호화 할 수 있습니다.

- 프로그램명 : highcopy
- 사용방법 1 : highcopy 원본파일명 복사본파일명
 - 예: highcopy original.txt new.txt 라고 입력하면 원본파일인 original.txt 파일과 동일한 사본을 생성하고, 사본의 이름은 new.txt 가 됩니다.
- 사용방법 2 : highcopy -d 숫자 원본파일명 복사본파일명

예: highcopy -d 7 original.txt new.txt 라고 입력하면 original.txt 파일에서 읽은 모든 바이트에 각각 7을 뺀 뒤, 새로운 파일 new.txt 에 기록함.

[문제]

직전 문제에서 사용한 alphabet.txt 파일을 사용합니다.

[1] highcopy alphabet.txt high_alpha.txt 입력하세요. 다음으로, cat high_alpha.txt 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[2] highcopy -d 7 crypt_alpha.txt plain_alpha.txt 입력하세요. 다음으로, cat plain_alpha.txt 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

소스코드도 제출해야 합니다.

답변 [1]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q02$ ./highcopy alphabet.txt high_alpha.txt
yeongju@vm-ubuntu20:~/sp2021-2/w04/q02$ cat high_alpha.txt
abcdefghijklmnopqrstuvwxyz
```

답변 [2]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q02$ ./highcopy -d 7 crypt_alpha.txt plain_alpha.txt
yeongju@vm-ubuntu20:~/sp2021-2/w04/q02$ cat plain_alpha.txt
abcdefghijklmnopqrstuvwxyz
```

[Q 3] 파일 오프셋 [배점: 30]

주어진 파일에서 지정한 위치로 오프셋을 옮긴 뒤, 정해진 개수만큼의 바이트(char)를 순방향 또는 역방향으로 읽어서 터미널에 출력하는 프로그램을 작성하세요

- 프로그램명 : seeker
- 사용방법 : seeker 파일명 시작위치 바이트수 (참고: 시작위치와 바이트수는 정수)
 - 예: seeker file.txt 10 5 입력하면, file.txt 파일에서 오프셋을 10 만큼 순방향으로 이동한 뒤(파일의 시작 위치를 기준으로, 파일의 끝 방향으로 10 만큼 이동함), 5 개의 바이트를 순방향으로(= 파일의 끝 방향으로) 읽어서 터미널에 출력
 - 예: seeker file.txt -5 6 라고 입력하면, file.txt 파일의 끝에서 오프셋을 **역방향으로(= 파일의 시작 위치 방향으로)** 5 만큼 이동한 뒤, 6 개의 바이트를 순방향으로 읽어서 터미널에 출력
 - 예: seeker file.txt begin 3 라고 입력하면, file.txt 파일의 처음으로 이동한 뒤, 3 개의 바이트를 순방향으로 읽어서 터미널에 출력.
 - 예: seeker file.txt end 3 라고 입력하면, file.txt 파일의 끝으로 이동한 뒤, 3 개의 바이트를 순방향으로 읽어서 터미널에 출력. (이 경우, seeker file.txt begin 3 과 동일한 결과가 나옵니다.)
 - 예: seeker file.txt mid 3 라고 입력하면, file.txt 파일의 중간으로 이동한 뒤, 3 개의 바이트를 순방향으로 읽어서 터미널에 출력. 파일에 “ab”가 저장된 경우 중간은 'a' 다음이고, 파일에 “abc”가 저장된 경우 중간은 'b' 다음이다.
 - 순방향으로 읽는 경우: 읽어야 할 바이트 수를 다 읽지 못하고 파일의 끝을 만나는

경우, 파일의 처음으로 돌아가서 순방향으로 읽기.

- ◆ 예: “abc”가 저장된 file.txt 파일에서 `seeker file.txt begin 5` 를 입력하면 `abcb` 를 출력

[문제]

직전 문제에서 사용한 `alphabet.txt` 파일을 사용합니다.

[1] `seeker alphabet.txt 0 5` 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[2] `seeker alphabet.txt begin 52` 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[3] `seeker alphabet.txt mid 13` 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[4] `seeker alphabet.txt end 52` 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[5] `seeker alphabet.txt 10 26` 입력 후, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

소스코드도 제출해야 합니다. (저수준 또는 고수준 파일 입출력 중, 본인이 원하는 방식으로 코딩하면 됩니다.)

답변 [1]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q03$ ./seeker alphabet.txt 0 5
abcb
```

답변 [2]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q03$ ./seeker alphabet.txt begin 52
abcdefghijklmnpqrstuvwxyz
```

답변 [3]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q03$ ./seeker alphabet.txt mid 13
nopqrstuvwxyz
```

답변 [4]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q03$ ./seeker alphabet.txt end 52  
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
```

답변 [5]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w04/q03$ ./seeker alphabet.txt 10 26  
klmnopqrstuvwxyzabcdefghij
```

끝! 수고하셨습니다 ☺