

[506489] 시스템프로그래밍**실습 #02 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 실습 시간 전날까지(일요일 까지)**
 - **수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)**
 - 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 리눅스 활용 및 셸 프로그램에 대한 내용입니다. 우분투 리눅스에서 패키지 설치, 기본 명령어 사용 및 텍스트 에디터 사용에 관한 문제와 함께 간단한 셸 스크립트를 작성하는 문제로 구성되어 있습니다.

패키지를 설치하는 과제를 할 때, 관리자 권한이 있는 경우(예: 가상 머신) `$sudo` 명령을 사용하면 됩니다. 관리자 권한이 없는 경우에는(예: 우분투 서버) 도커를 사용하여 우분투 컨테이너를 사용하면 관리자 권한을 획득할 수 있습니다. 도커 사용과 관련된 내용은 강의노트 “도커를 사용하여 우분투 컨테이너 사용” 부분을 참고하세요.

<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 vim 에디터 사용법과 셸 스크립트 사용법을 익힐 수 있었습니다.

또한, 기본적으로 많이 사용하는 명령어(pwd, cd, ls 등)를 익힐 수 있었습니다.

[Q 1] 디렉토리 관리 [배점: 10]

HOME 은 환경변수로써, 각 계정의 홈 디렉토리를 저장하고 있습니다. `$echo $HOME` 이라는 명령을 입력하면 `/home/daniel` 과 같이, 현재 로그인한 계정의 홈 디렉토리를 출력합니다. `$cd $HOME` 을 입력하면 계정의 홈 디렉토리로 이동합니다. 앞으로 **`$HOME` 은 계정의 홈 디렉토리를 의미하는 것으로 사용**하겠습니다.

다음과 같이 디렉토리를 생성하세요: `$HOME/sp2021-2/w02/` (w02 는 week 02 를 의미합니다). 최 하위 디렉토리 w02 로 이동한 후, `$pwd` 명령을 입력하세요. 터미널 출력 결과를 캡처해서 첨부하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w02$ pwd
/home/yeongju/sp2021-2/w02
```

[Q 2] 리눅스 명령어 [배점: 10]

리눅스 명령어 중, 최근에 입력한/사용한 명령어를 모두 터미널에 출력하는 명령어가 있습니다. 1) 어떤 명령어 인가요?

2) 가장 최근에 사용한 10 개의 명령어만 보려면 합니다. 터미널에 어떻게 입력해야 하는지 답하세요.

3) 최근에 사용한 명령어 내역을 초기화하려고 합니다. 터미널에 어떻게 입력해야 하는지 답하세요.

답변 1): `$history`

답변 2): `$history 10`

답변 3): \$history -c

[Q 3] 리다이렉션 (redirection) [배점: 10]

터미널에서 \$ls -al 을 입력하세요.

1) 터미널 출력 결과를 보고, \$ls -al 명령이 어떤 역할을 하는지 설명하세요.

터미널에서 \$ls -al > list.txt 를 입력하세요.

2) 'list.txt' 파일을 열어서 내용을 확인하고, '>' 연산자가 어떤 역할을 하는지 설명하세요.

이 상태에서 \$pwd > list.txt 를 입력하세요. 'list.txt' 파일을 열어서 내용을 확인하세요. 이 상태에서 \$pwd >> list.txt 를 입력하세요. 'list.txt' 파일을 열어 확인하세요.

3) > 연산자와 >> 연산자의 차이점은 무엇인가요??

답변 1): 숨긴 파일을 포함하여 모든 파일 및 디렉토리 목록을 상세하게 출력한다.

답변 2): 덮어쓰기 같은 역할을 한다. (list.txt 파일을 생성하고 \$ls -al 명령의 출력 내용을 파일에 쓰기)

답변 3): > 연산자는 새로운 내용을 덮어쓴다면, >> 연산자는 기존 내용에 이어쓴다.

[Q 4] vim 에디터 [배점: 10]

1) vim 에디터에서 라인 번호가 표시되도록 설정하세요. 화면을 캡처해서 첨부하세요

2) vim 에디터에서 화면을 위/아래로 분할해 보세요. 화면을 캡처해서 첨부하세요.

3) vim 에디터에서 탭 페이지를 만들어보세요(= 여러 개의 탭을 만들어서, 여러 개의 소스코드/텍스트파일을 하나의 터미널에서 open 하는 것). 화면을 캡처해서 첨부하세요.

4) vim 에디터를 이용해서 두개의 파일(file1.txt, file2.txt)을 비교하려 합니다 (두 파일의 차이점 비교/확인). 터미널에서 어떻게 입력해야 할까요?

답변 1):

```
yeongju@vm-ubuntu20: ~/sp2021-2/w02
```

```
1 Hello World!  
2 Good Bye..  
3  
4  
5  
6  
7  
8 end.
```

```
:set number
```

```
8,3 All
```

답변 2):

```

1 Hello World!
2 Good Bye..
3
4
5
6
7
8 en.

```

yeongju@vm-ubuntu20: ~/sp2021-2/w02

[No Name] [+]

8,3 All

답변 3):

A screenshot of a terminal window titled "yeongju@vm-ubuntu20: ~/sp2021-2/w02". The terminal has a dark purple background. At the top, there's a tab bar with three tabs labeled "[No Name]", "[No Name]", and "tt", followed by a search icon, a menu icon, and window control icons. Below the tab bar, the prompt shows eight numbered lines:
1 Hello World!
2 Good Bye..
3
4
5
6
7
8 end.
The cursor is positioned at the end of line 8. On the left side of the terminal, there are several tilde (~) symbols representing scrollback history. At the bottom right, there are status indicators: "d", "8,1", and "All".

답변 4): \$vim -d file1.txt file2.txt

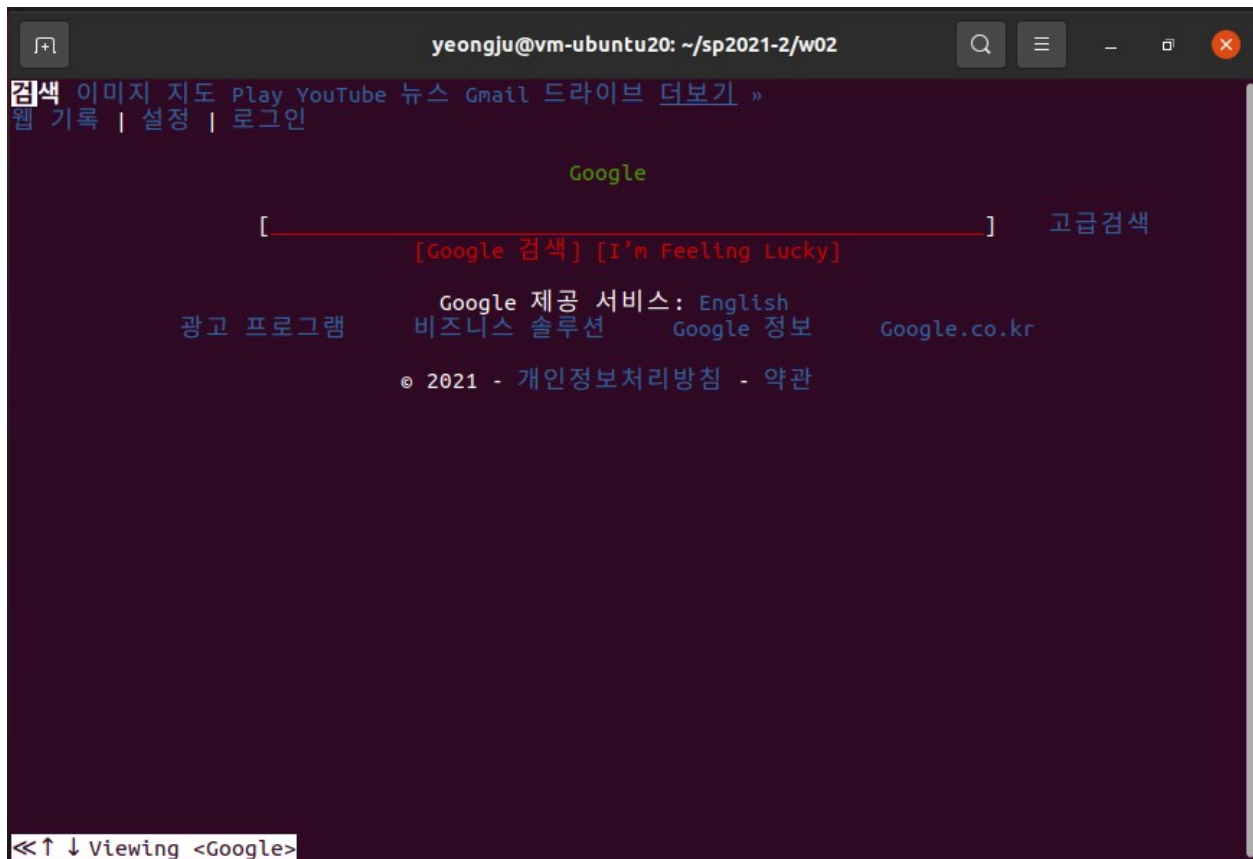
[Q 5] 패키지 관리 [배점: 10]

리눅스에는 w3m 이라는 프로그램이 있습니다. 1) w3m 은 무엇인가요?

w3m을 설치하세요. 2) w3m을 구동한 화면을 캡처해서 본 문서에 첨부하세요.

답변 1): w3m은 무료 오픈소스 텍스트 기반 웹 브라우저다.

답변 2):



[Q 6] 셸 프로그래밍 [배점: 20]

아래와 같이 동작하는 fibo.sh 라는 이름을 가진 셸 스크립트를 작성하세요.

- 사용자로부터 숫자 한 개를 입력 받음 (숫자는 0 보다 큰 정수라고 가정)
- 사용자가 n 을 입력했을 때, n 번째 Fibonacci 숫자를 계산 후 출력

재귀호출을 사용해도 되고 반복문을 이용해서 구현해도 됩니다. 스크립트를 실행하고 숫자를 입력해서 테스트 하세요. 터미널 화면을 캡처해서 첨부하세요. 셸 스크립트 소스코드를 아래에 붙여 넣으세요..

답변 (터미널 화면 캡처):


```
yeongju@vm-ubuntu20: ~/sp2021-2/w02$ ./fibonacci.sh
Enter a number: 1
result: 0
yeongju@vm-ubuntu20: ~/sp2021-2/w02$ ./fibonacci.sh
Enter a number: 2
result: 1
yeongju@vm-ubuntu20: ~/sp2021-2/w02$ ./fibonacci.sh
Enter a number: 3
result: 1
yeongju@vm-ubuntu20: ~/sp2021-2/w02$ ./fibonacci.sh
Enter a number: 4
result: 2
yeongju@vm-ubuntu20: ~/sp2021-2/w02$ ./fibonacci.sh
Enter a number: 5
result: 3
yeongju@vm-ubuntu20: ~/sp2021-2/w02$
```

답변 (소스코드):

```
#!/bin/bash
```

```
echo -n "Enter a number: "
```

```
read num
```

```
x=0
```

```
y=1
```

```
temp=0
```

```
if [ $num -eq 1 ]
then
    echo "result: 0"
elif [ $num -eq 2 ]
then
    echo "result: 1"
else
    for (( i=1; i<$(expr $num - 1); i++ ))
    do
        temp=$(expr $x + $y)
        x=$y
        y=$temp
    done

    echo "result: $y"
fi
```

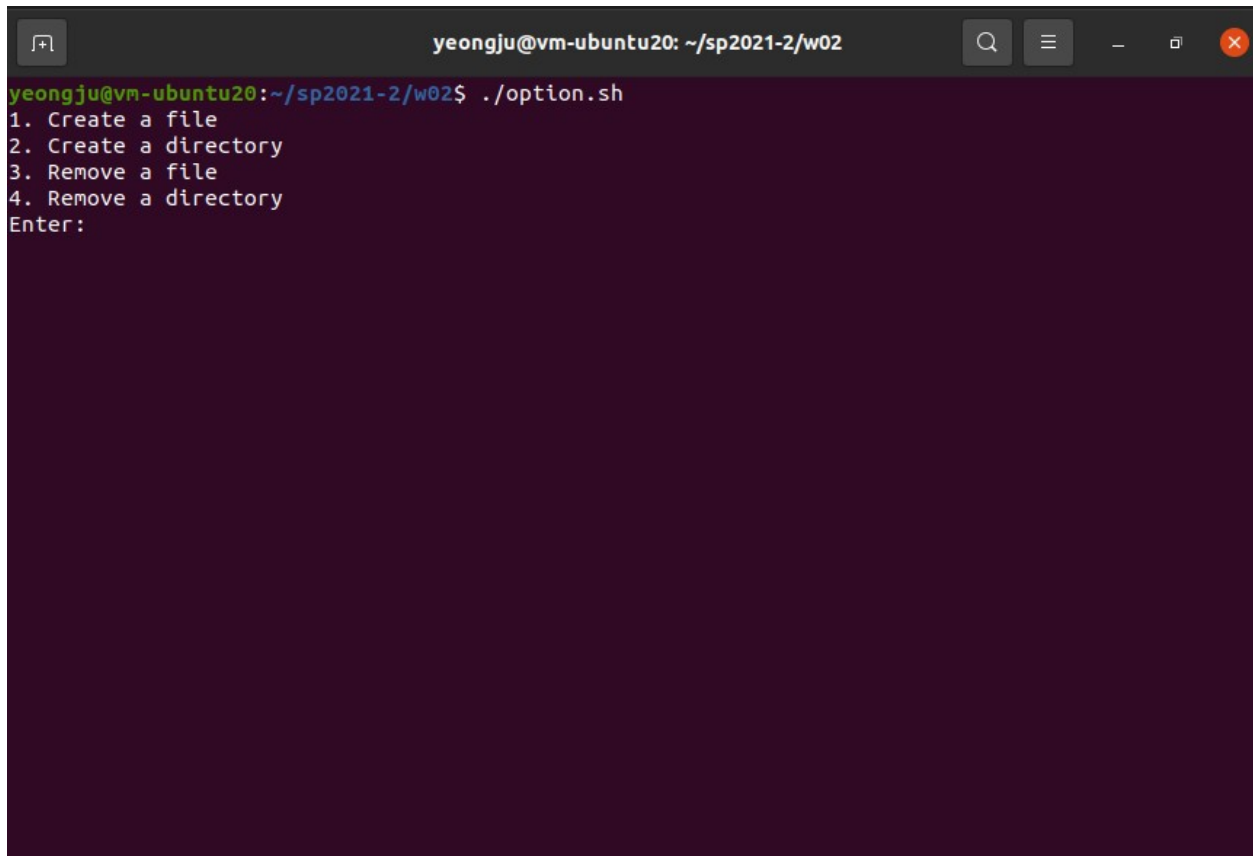
[Q 7] 셸 프로그래밍 [배점: 20]

아래를 참고하여 option.sh 라는 이름의 셸 스크립트를 작성하세요.
스크립트 실행 시, 아래와 같이 출력하고 사용자의 입력을 기다립니다.

1. Create a file
 2. Create a directory
 3. Remove a file
 4. Remove a directory
- Enter:

사용자가 1 또는 3을 선택하면, **Enter filename:** 을 출력하고 사용자의 입력을 기다리고, 사용자가 2 또는 4를 선택하면 **Enter directory name:** 을 출력하고 사용자의 입력을 기다립니다. 사용자의 입력에 따라 파일/디렉토리를 생성/삭제하고 종료하는 **셸 스크립트**를 작성하세요. 소스코드 내용을 아래에 붙여 넣으세요. 프로그램을 실행하고, 첫 화면 (=메뉴가 출력된 화면)을 캡처해서 첨부하세요

답변 (첫 화면 캡처):

A terminal window titled 'yeongju@vm-ubuntu20: ~/sp2021-2/w02'. The prompt is 'yeongju@vm-ubuntu20:~/sp2021-2/w02\$'. The user has entered './option.sh'. The script output is: '1. Create a file', '2. Create a directory', '3. Remove a file', '4. Remove a directory', and 'Enter:'. The terminal background is dark purple.

```
yeongju@vm-ubuntu20: ~/sp2021-2/w02
yeongju@vm-ubuntu20:~/sp2021-2/w02$ ./option.sh
1. Create a file
2. Create a directory
3. Remove a file
4. Remove a directory
Enter:
```

답변 (소스코드):

```
#!/bin/bash
```

```
create_and_remove()
```

```
{
```

```
    if [ $1 -eq 1 ]
    then
        touch $2
    elif [ $1 -eq 2 ]
    then
        mkdir $2
    elif [ $1 -eq 3 ]
    then
        rm $2
    else
        rm -r $2
    fi
}

echo "1. Create a file"
echo "2. Create a directory"
echo "3. Remove a file"
echo "4. Remove a directory"

echo -n "Enter: "
read num

name=""
```

```
if [ $num -eq 1 ] || [ $num -eq 3 ]
then
    echo -n "Enter filename: "
elif [ $num -eq 2 ] || [ $num -eq 4 ]
then
    echo -n "Enter directory name: "
else
    echo "not a number"
    exit 0
fi

read name
create_and_remove $num $name
```

끝! 수고하셨습니다 ☺