

## 4주차 결과보고서

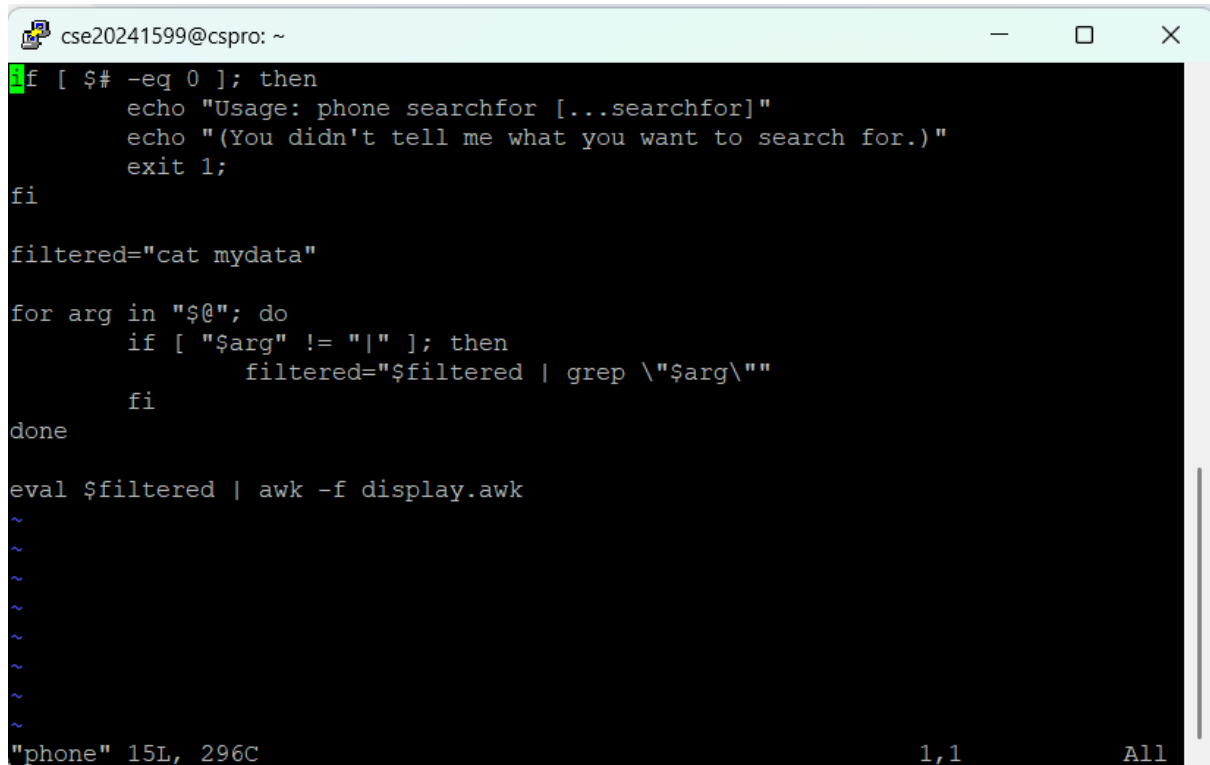
전공: 컴퓨터공학과

학년: 2학년

학번: 20241599

이름: 박은석

1.



```
cse20241599@cspro: ~  
if [ $# -eq 0 ]; then  
    echo "Usage: phone searchfor [...searchfor]"  
    echo "(You didn't tell me what you want to search for.)"  
    exit 1;  
fi  
  
filtered="cat mydata"  
  
for arg in "$@"; do  
    if [ "$arg" != "|" ]; then  
        filtered="$filtered | grep \"$arg\""  
    fi  
done  
  
eval $filtered | awk -f display.awk  
~  
~  
~  
~  
~  
~  
~  
"phone" 15L, 296C 1,1 All
```

```
if [ $# -eq 0 ]; then
```

```
    echo "Usage: phone searchfor [...searchfor]"
```

```
    echo "(You don't tell me what you want to search for.)"
```

```
    exit 1;
```

```
fi
```

: 입력된 인자가 없으면 에러 메시지를 출력하고 프로그램을 종료하는 코드

```
filtered="cat mydata"
```

: mydata 파일에 있는 텍스트를 filtered 변수에 그대로 저장한다.

```

for arg in "$@"; do
    if [ "$arg" != "|" ]; then
        filtered="$filtered | grep -w"$arg""
    fi
done

```

: 입력된 문자열을 arg 변수에 저장하고, arg가 | 가 아니라면 filtered에서 '| grep "arg"'가 추가된다. 이 과정을 입력된 문자열만큼 반복한다. 예를 들어, './phone 홍길동 서울시'가 입력되었으면 for문 전 filtered="cat mydata"이고, for문을 시작하면 filtered="cat mydata | grep "홍길동"이 되고, 다시 반복하여 filtered="cat mydata | grep '홍길동' | grep '서울시'가 된다.

grep 명령어는 특정 파일에서 문자열을 포함한 행을 출력한다. '홍길동'을 포함한 행이 '서울시'도 포함한다면 출력이 되는 방식이기 때문에, and 형식으로 검색하기에 적합하다.

```
eval filtered | awk -f display.awk
```

: eval은 문자열을 명령문으로 인식하고 실행하는 명령어이다. filtered 변수의 cat, grep의 명령어를 실행할 수 있다. eval 명령이 다 끝난 후 awk로 결과를 넘겨서 display.awk 파일의 명령을 실행하여 이름, 주소, 전화번호를 출력할 수 있다.