**오픈 소스 기초 설계 [쉘 스크립트 과제 제출 보고서]**

AI융합학부 20231764 박시우

ex3-0.sh

* Hello World 출력

[결과화면]



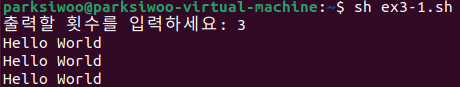
[코드]

echo "Hello World"

ex3-1.sh

* 숫자를 입력 받아 Hello World를 여러번 출력

[결과화면]



[코드]

read -p "출력할 횟수를 입력하세요: " count

for i in $(seq $count)

do

echo "Hello World"

done

ex3-2.sh

* 두 숫자와 연산자(+ 또는 -)를 입력 받아 계산하기.

[결과화면]



[코드]

echo "수식을 입력하세요 (예: 3 - 1):"

read expression

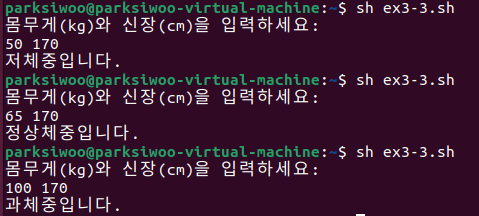
result=$(echo "$expression" | bc)

echo "결과: $result"

ex3-3.sh

* 체질량 지수 (BMI) 계산하여 비만 여부를 판단.

[결과화면]



[코드]

echo "몸무게(kg)와 신장(cm)을 입력하세요:"

read weight height

# BMI 계산

bmi=$(awk -v weight=$weight -v height=$height 'BEGIN { printf "%.2f\n", weight / ((height / 100) \* (height / 100)) }')

# BMI에 따라 체중 상태 판단

case 1 in

$(echo "$bmi < 18.5" | bc -l)) echo "저체중입니다." ;;

$(echo "$bmi >= 18.5 && $bmi < 23" | bc -l)) echo "정상체중입니다." ;;

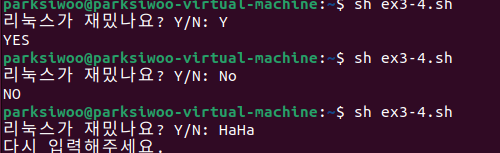
\*) echo "과체중입니다." ;;

esac

ex3-4.sh

* case문을 활용한 예제 실습
* 입력에 따라 분기하여 서로 다른 출력. 다양한 입력에 대한 방어 코드.

[결과화면]



[코드]

read -p "리눅스가 재밌나요? Y/N: " response

case "$response" in

[Yy])

echo "YES"

;;

[Nn][Oo]\*)

echo "NO"

;;

\*)

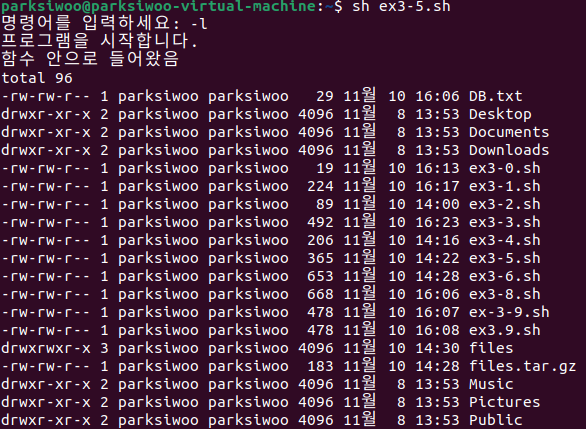
echo "다시 입력해주세요."

;;

esac

ex3-5.sh

* 내부 함수를 만들어 리눅스 명령어(e.g. ls)를 실행
* 입력 인자를 내부 함수 내의 리눅스 명령어 옵션으로 전달

[결과화면]

[코드]

run\_ls() {

# 입력된 인자로 ls 명령어 실행

ls "$@"

}

# 사용자로부터 인자를 입력 받아서 함수 호출

echo "리눅스 명령어를 실행합니다."

read -p "옵션을 포함한 명령어를 입력하세요: " user\_input

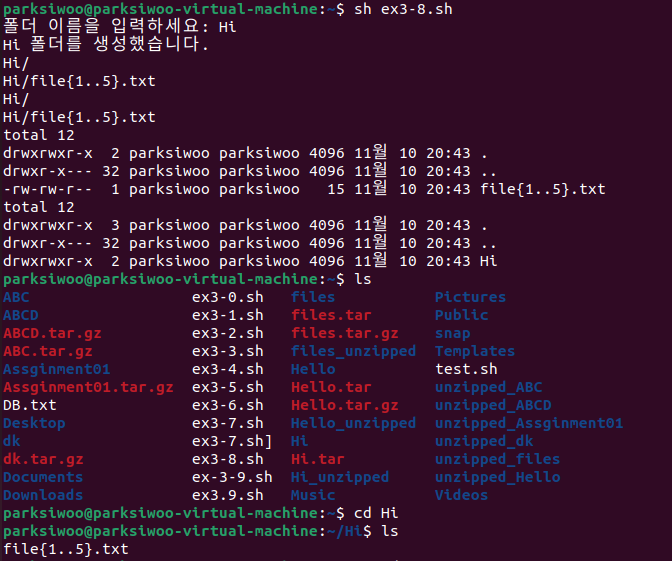
# 입력된 명령어를 함수로 전달

run\_ls $user\_input

ex3-6.sh

* 입력된 이름으로 폴더가 있는 확인. 없다면 생성.
* 5개의 파일을 만들고 압축. 새로운 폴더를 생성하여 압축 해제

[결과화면]



[코드]

read -p "폴더 이름을 입력하세요: " folder\_name

if [ ! -d "$folder\_name" ]; then

mkdir "$folder\_name"

echo "$folder\_name 폴더를 생성했습니다."

else

echo "$folder\_name 폴더가 이미 존재합니다."

fi

for i in {1..5}; do

echo "file$i.txt" > "$folder\_name/file$i.txt"

done

tar -cvf "$folder\_name.tar" "$folder\_name"

new\_folder="${folder\_name}\_unzipped"

mkdir "$new\_folder"

tar -xvf "$folder\_name.tar" -C "$new\_folder"

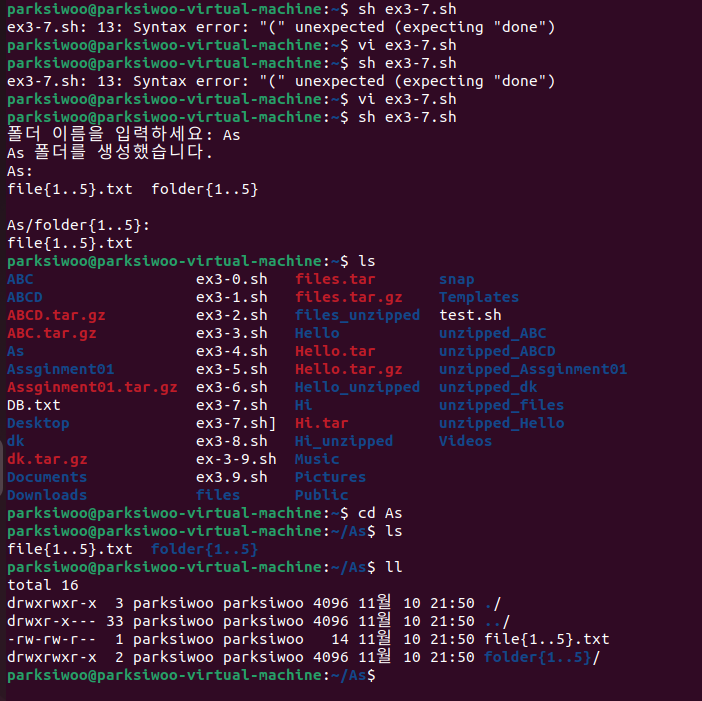
ls -al "$folder\_name"

ls -al "$new\_folder"

ex3-7.sh

* 입력된 이름으로 폴더에서 5개 이상의 파일을 만듦.
* 파일 이름대로 하위 폴더들을 생성하고 각 폴더에 해당 파일을 링크.

[결과화면]



[코드]

read -p "폴더 이름을 입력하세요: " folder\_name

if [ ! -d "$folder\_name" ]; then

mkdir "$folder\_name"

echo "$folder\_name 폴더를 생성했습니다."

else

echo "$folder\_name 폴더가 이미 존재합니다."

fi

for i in {1..5}; do

echo "내용 $i" > "$folder\_name/file$i.txt"

done

for i in {1..5}; do

subfolder="$folder\_name/folder$i"

mkdir "$subfolder"

ln -s "../file$i.txt" "$subfolder/file$i.txt"

done

ls -R "$folder\_name"

ex3-8.sh

* 팀원의 이름과 생일 또는 전화번호를 DB.txt에 기록
* 이름과 정보를 DB.txt에 계속 추가할 수 있음. 또는 처음부터 다시 작성

[결과화면]



[코드]

database="DB.txt"

# DB 파일이 존재하지 않으면 생성

if [ ! -e "$database" ]; then

touch "$database"

echo "DB 파일을 생성했습니다."

fi

while true; do

read -p "이름을 입력하세요 (종료하려면 'exit' 입력): " name

# 'exit' 입력 시 종료

if [ "$name" == "exit" ]; then

echo "프로그램을 종료합니다."

break

fi

read -p "생일 또는 전화번호를 입력하세요: " info

# 정보를 DB 파일에 추가

echo "$name: $info" >> "$database"

echo "DB에 추가되었습니다."

# DB.txt 파일 내용 출력

echo "DB.txt 파일 내용:"

cat "$database"

# 이름으로 검색하여 팀원 정보 확인

echo "검색된 팀원 정보:"

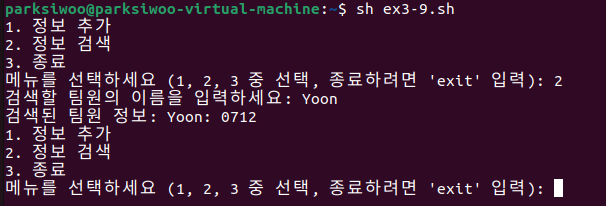
grep "$name" "$database"

done

ex3-9 .sh

* 이름으로 검색. DB.txt에서 검색된 팀원의 정보를 확인

[결과화면]



[코드]

#!/bin/bash

database="DB.txt"

# DB 파일이 존재하지 않으면 생성

if [ ! -e "$database" ]; then

touch "$database"

echo "DB 파일을 생성했습니다."

fi

# 이름으로 정보 검색 함수

search\_info() {

read -p "검색할 팀원의 이름을 입력하세요: " search\_name

# DB.txt 파일에서 이름으로 검색하여 결과 출력

search\_result=$(grep -i "$search\_name" "$database")

if [ -n "$search\_result" ]; then

echo "검색된 팀원 정보: $search\_result"

else

echo "검색된 팀원이 없습니다."

fi

}

while true; do

echo "1. 정보 추가"

echo "2. 정보 검색"

echo "3. 종료"

read -p "메뉴를 선택하세요 (1, 2, 3 중 선택, 종료하려면 'exit' 입력): " choice

case $choice in

1)

read -p "이름을 입력하세요: " name

read -p "생일 또는 전화번호를 입력하세요: " info

# 정보를 DB 파일에 추가

echo "$name: $info" >> "$database"

echo "DB에 추가되었습니다."

;;

2)

search\_info

;;

3 | "exit")

echo "프로그램을 종료합니다."

exit 0

;;

\*)

echo "잘못된 선택입니다. 1, 2, 또는 3 중에서 선택하세요."

;;

esac

done