오픈소스 기초설계 셀 스크립트

이도훈 20233099

ex3-0.sh

#!/bin/bash

echo "Hello, World!"



#!/bin/bash

# 사용자로부터 숫자 입력 받기

read -p "출력할 횟수를 입력하세요 (무한 루프를 위해 Enter 키만 누르세요): " num

if [ -z "$num" ]; then

# 무한 루프로 "Hello, World!" 출력

while true; do

echo "Hello, World!"

done

else

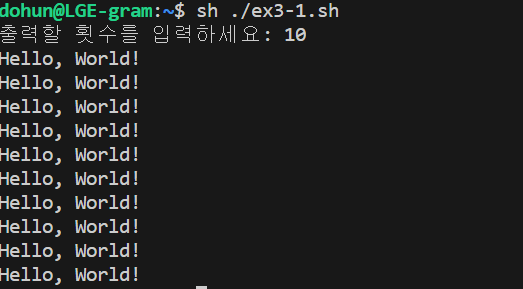
# 입력된 숫자만큼 "Hello, World!" 출력

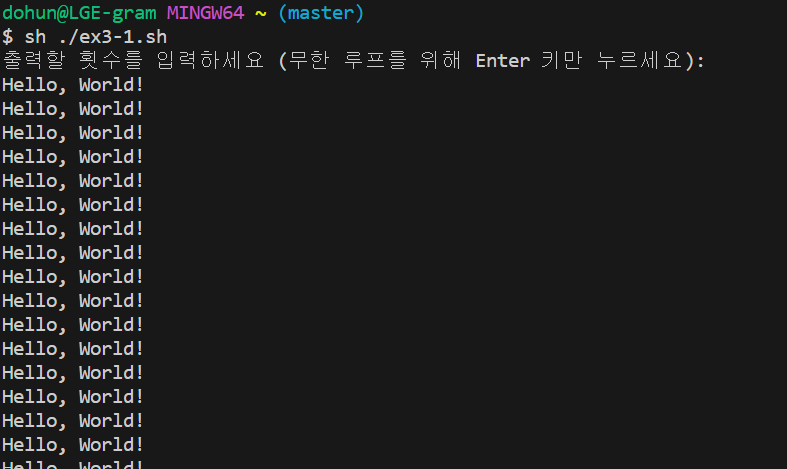
for ((i=1; i<=num; i++)); do

echo "Hello, World!"

done

fi





ex3-2.sh

#!/bin/bash

# 사용자로부터 두 숫자 입력 받기

read -p "첫 번째 숫자를 입력하세요: " num1

read -p "두 번째 숫자를 입력하세요: " num2

# 사용자로부터 연산자 입력 받기

read -p "덧셈 (+) 또는 뺄셈 (-) 연산을 선택하세요: " operator

# 입력된 연산자에 따라 계산 수행

if [ "$operator" = "+" ]; then

result=$((num1 + num2))

operator\_text="덧셈"

elif [ "$operator" = "-" ]; then

result=$((num1 - num2))

operator\_text="뺄셈"

else

echo "유효하지 않은 연산자입니다."

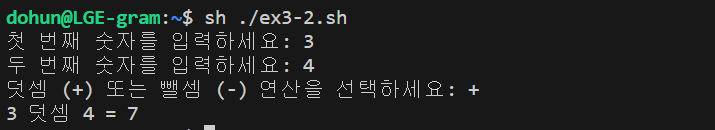
exit 1

fi

exit 0

# 결과 출력

echo "$num1 $operator\_text $num2 = $result"



#!/bin/bash

# 사용자로부터 몸무게(kg) 입력 받기

read -p "몸무게(kg)를 입력하세요: " weight

# 사용자로부터 신장(m) 입력 받기

read -p "신장(m)를 입력하세요: " height

# BMI 계산

bmi=$(echo "scale=2; $weight / ($height \* $height)" | bc -l)

# BMI 결과 출력

echo "계산된 BMI: $bmi"

# 비만 여부 판단

if (( $(echo "$bmi >= 18.5 && $bmi < 25" | bc -l) )); then

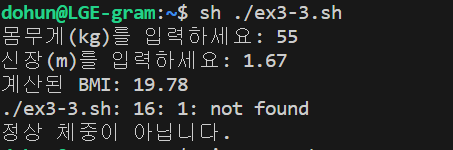
echo "정상 체중입니다."

else

echo "정상 체중이 아닙니다."

fi

exit 0



ex3-4.sh

#!/bin/bash

# 사용자로부터 "yes" 또는 "no" 입력 받기 (대소문자 구분 없이)

read -p "리눅스가 재밌나요? (yes / no): " response

# 입력값을 모두 소문자로 변환

response\_lower=$(echo "$response" | tr '[:upper:]' '[:lower:]')

# 입력값이 유효한지 확인

if [[ "$response\_lower" == \*yes\* || "$response\_lower" == "y" ]]; then

echo "멋져요! 리눅스는 정말 재밌습니다."

elif [[ "$response\_lower" == \*no\* || "$response\_lower" == "n" ]]; then

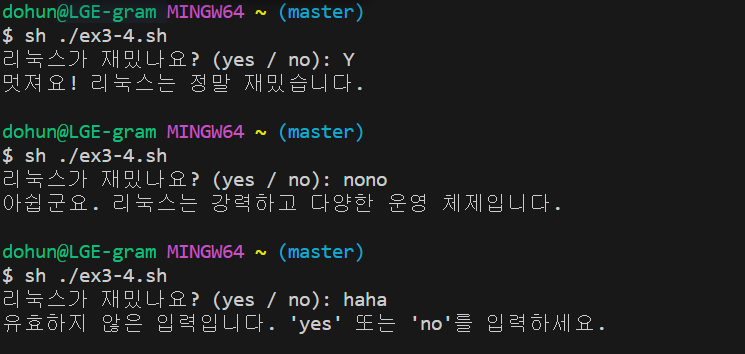
echo "아쉽군요. 리눅스는 강력하고 다양한 운영 체제입니다."

else

echo "유효하지 않은 입력입니다. 'yes' 또는 'no'를 입력하세요."

fi

exit 0



#!/bin/bash

# 내부 함수로 리눅스 명령어 실행

run\_linux\_command() {

# 사용법을 표시하는 도움말 메시지

help\_message="사용법: $0 <command> [options]\n예: $0 ls -l"

# 명령어가 전달되지 않았을 경우 도움말 메시지 출력

if [ $# -eq 0 ]; then

echo -e "명령어를 지정하세요.\n$help\_message"

return

fi

# 첫 번째 인자로 받은 명령어를 실행

command="$1"

shift # 첫 번째 인자 삭제

# 명령어가 존재하는지 확인

if ! command -v "$command" &> /dev/null; then

echo "오류: '$command' 명령어를 찾을 수 없습니다."

return

fi

# 명령어 실행

"$command" "$@"

}

# 사용법 표시

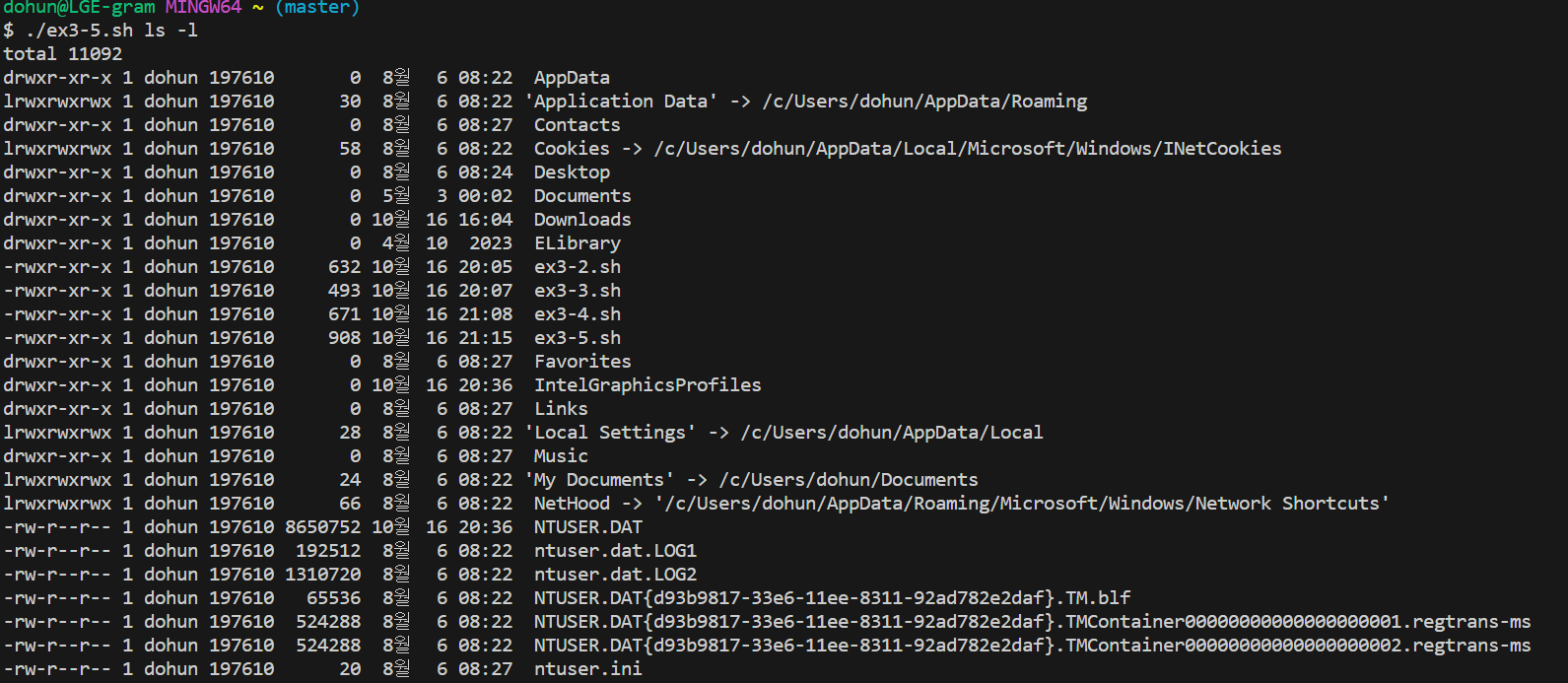
if [ $# -eq 0 ]; then

echo "내부 함수로 리눅스 명령어 실행"

echo "사용법: $0 <command> [options]"

echo "예: $0 ls -l"

exit

fi

ex3-6.sh

#!/bin/bash

# 사용자로부터 폴더 이름 입력 받기

read -p "폴더 이름을 입력하세요: " folder\_name

# 폴더가 존재하지 않으면 생성

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

mkdir "$folder\_name"

echo "폴더 '$folder\_name'를 생성했습니다."

fi

# 5개의 파일 생성 (예를 들어, 파일1.txt, 파일2.txt, ...)

for i in {1..5}; do

touch "$folder\_name/파일$i.txt"

echo "파일$i.txt를 생성했습니다."

done

# 파일들을 압축

tar -czvf "$folder\_name.tar.gz" "$folder\_name"

echo "파일들을 압축했습니다: $folder\_name.tar.gz"

# 새로운 폴더를 생성하여 압축 해제

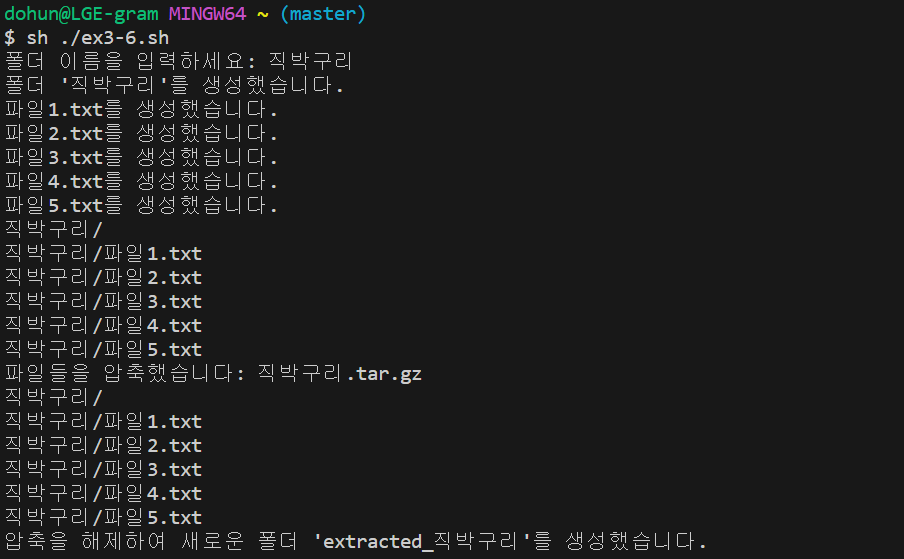
new\_folder="extracted\_$folder\_name"

mkdir "$new\_folder"

tar -xzvf "$folder\_name.tar.gz" -C "$new\_folder"

echo "압축을 해제하여 새로운 폴더 '$new\_folder'를 생성했습니다."

exit 0



ex3-7.sh

#!/bin/bash

# 사용자로부터 폴더 이름 입력 받기

read -p "폴더 이름을 입력하세요: " folder\_name

# 폴더가 존재하지 않으면 생성

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

mkdir "$folder\_name"

echo "폴더 '$folder\_name'를 생성했습니다."

fi

# 5개 이상의 파일 생성 (예를 들어, 파일1.txt, 파일2.txt, ...)

for i in {1..5}; do

touch "$folder\_name/파일$i.txt"

echo "파일$i.txt를 생성했습니다."

done

# 파일 이름대로 하위 폴더 생성하고 각 폴더에 파일을 링크

for i in {1..5}; do

subfolder\_name="하위폴더$i"

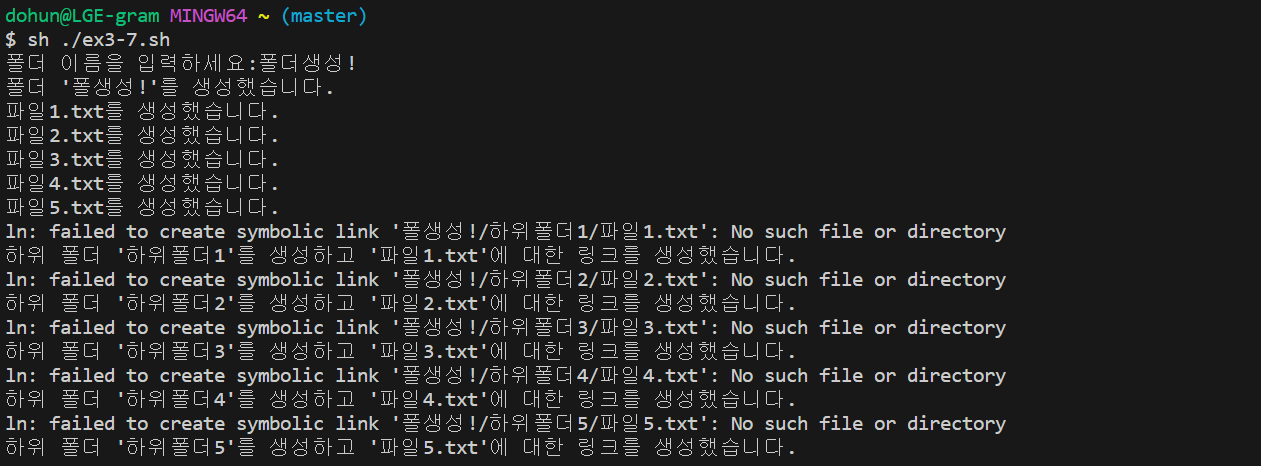
mkdir "$folder\_name/$subfolder\_name"

ln -s "../파일$i.txt" "$folder\_name/$subfolder\_name/파일$i.txt"

echo "하위 폴더 '$subfolder\_name'를 생성하고 '파일$i.txt'에 대한 링크를 생성했습니다."

done

exit 0



ex3-8.sh

# DB 파일명

db\_file="DB.txt"

# DB 파일이 존재하지 않으면 새로 작성

if [ ! -e "$db\_file" ]; then

touch "$db\_file"

fi

while true; do

clear

echo "1. 정보 추가"

echo "2. DB 초기화 (모든 데이터 삭제)"

echo "3. 나가기"

read -p "선택하실 작업을 입력하세요 (1/2/3): " choice

case $choice in

1)

# 정보 추가

read -p "팀원의 이름을 입력하세요: " name

read -p "팀원의 생일 또는 전화번호를 입력하세요: " info

echo "$name: $info" >> "$db\_file"

echo "정보가 DB.txt에 추가되었습니다."

;;

2)

# DB 초기화 (모든 데이터 삭제)

> "$db\_file"

echo "DB.txt의 모든 데이터가 삭제되었습니다."

;;

3)

# 나가기

echo "프로그램을 종료합니다."

exit

;;

\*)

echo "잘못된 선택입니다. 다시 입력하세요."

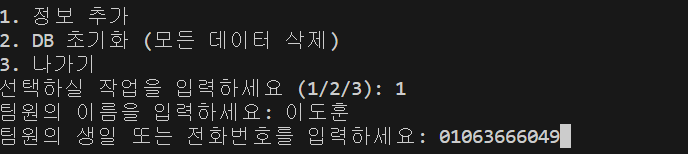
;;

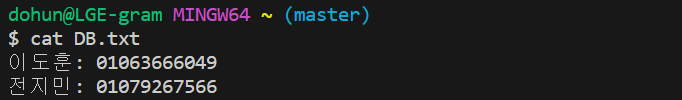
esac

read -p "계속하려면 Enter 키를 누르세요..."

done

exit 0





ex3-9.sh

#!/bin/bash

# DB 파일명

db\_file="DB.txt"

# DB 파일이 존재하지 않으면 새로 작성

if [ ! -e "$db\_file" ]; then

touch "$db\_file"

fi

if [ -z "$1" ]; then

echo "사용법: $0 <텍스트>"

exit 1

fi

search\_text="$1"

while IFS= read -r line; do

if [[ "$line" == \*"$search\_text"\* ]]; then

echo "검색 결과: $line"

fi

done < "$db\_file"

exit 0

