오픈소스 쉘 스크립트 보고서

학번 : 20231778 이름 : 이민혁

Ex3-0 코드

#! /bin/sh

echo "hello world"

exit 0

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Echo를 통해 hello world를 출력하였다.

Ex3-1

#! /bin/sh

number=$1

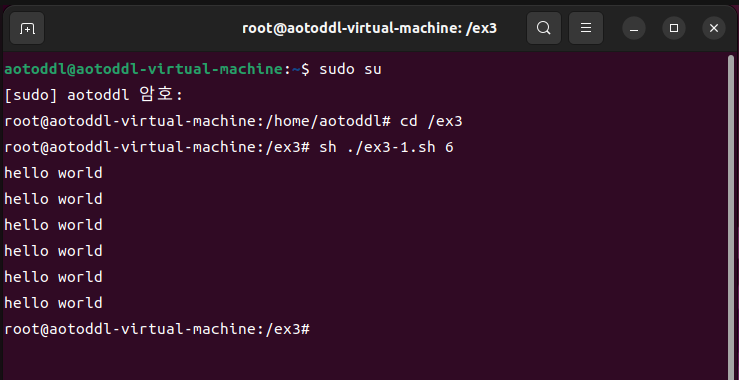
for i in $(seq 1 $number)

do

echo "hello world"

done

exit 0



반복할 숫자를 인자로 받고, for문과 echo를 이용하여 인자로 받은 숫자만큼 반복해서 출력한다.

Ex3-2

#!/bin/sh

number1=$1

operator=$2

number2=$3

if [ $operator = "+" ]

then

echo `expr $number1 + $number2`

elif [ $operator = "-" ]

then

echo `expr $number1 - $number2`

elif [ $operator = "/" ]

then

echo `expr $number1 / $number2`

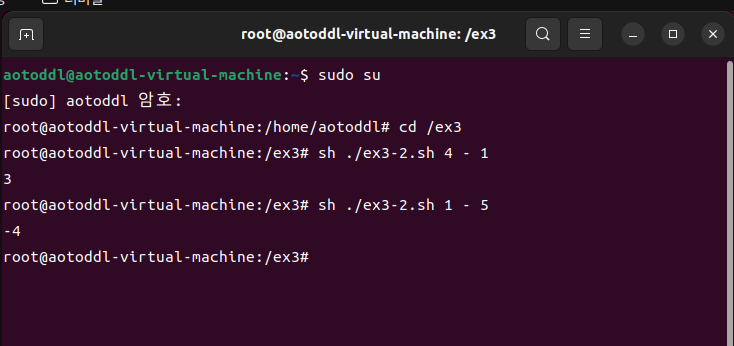
elif [ $operator = "\*" ]

then

echo `expr $number1 \\* $number2`

fi

exit 0



계산할 두 숫자와 연산자를 인자로 받은 후 if문을 통해 연산자를 구분한 뒤 expr을 이용해 연산하고, 그 값을 echo로 출력한다.

ex3-3

#!/bin/sh

weight=$1

height\_cm=$2

height\_m=$(echo " $height\_cm / 100" | bc -l)

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

if [ $(echo "$BMI < 18.5" | bc) -eq 1 ]

then

echo "저체중입니다."

elif [ $(echo "$BMI >= 18.5" | bc) -eq 1 ] && [ $(echo "$BMI < 23.5" | bc) -eq 1 ]

then

echo "정상체중입니다."

elif [ $(echo "$BMI >= 23" | bc) -eq 1 ]

then

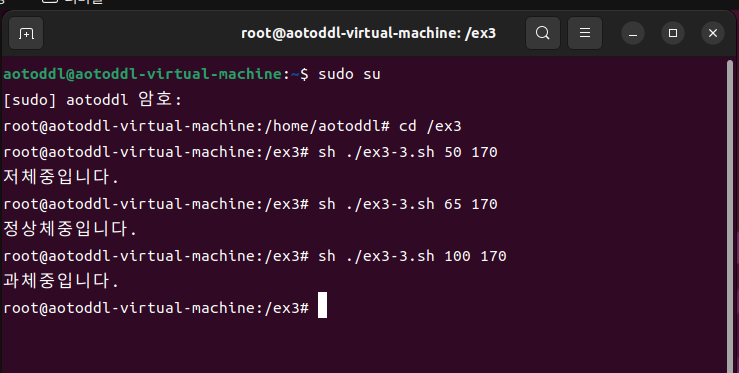
echo "과체중입니다."

else

echo "오류"

fi

exit 0



몸무게와 키를 인자로 받은 후 키를 m로 변환하고, bc를 통해 실수 계산을 하여 BMI를 구한 뒤 if문을 통해 검사한다.

if문에서 조건 검사를 할 때 -eq 1을 통해 bc를 이용한 실수 대소비교가 참인지 아닌지 구분한다.

그 후 조건에 따라서 과체중인지 정상체중인지 저체중인지를 echo로 출력한다.

Ex3-4

#!/bin/sh

echo "리눅스가 재미있나요? (yes / no)"

read mind

case "$mind" in

[yY]\*)

echo "yes";;

[nN]\*)

echo "no";;

\*)

echo "yes of no로 입력해 주세요.";;

esac

exit 0



질문에 대한 대답을 read로 받은 뒤 case를 통해서 답변받은 문구가 case 조건에 해당하는 문자로 시작하는지 검사한 뒤 Yy면 yes Nn면 no를 출력한다.

Ex3-5

#!/bin/sh

echo "프로그램을 시작합니다."

ls\_function() {

echo "함수 안으로 들어왔음"

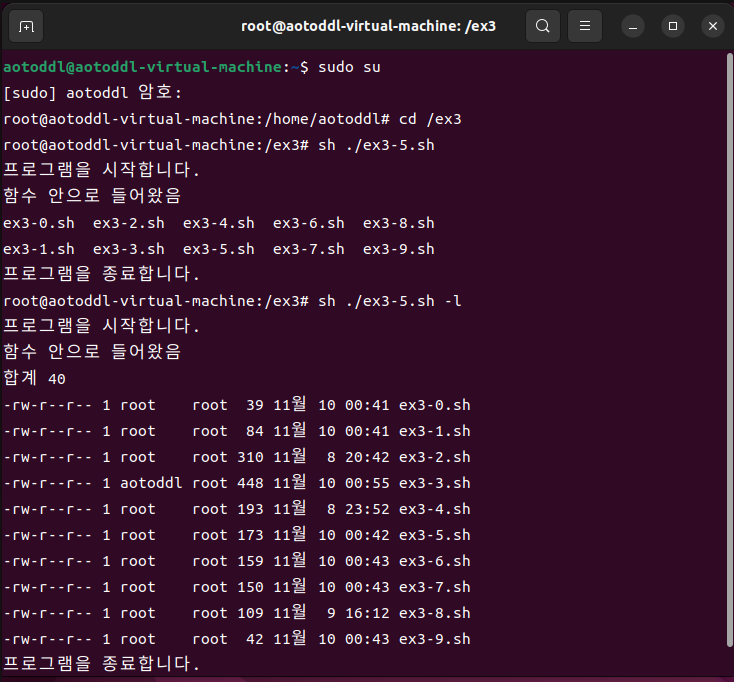
ls $1

}

ls\_function $1

echo "프로그램을 종료합니다."

exit 0



Echo를 통해 프로그램을 시작한다는 것을 출력한다. 함수를 만들고, 함수 내에서 echo를 통해 함수 안으로 들어왔다는 것을 출력한 뒤 ls 와 받은 인자를 더한 명령어를 실행한다. 함수가 끝나면 echo를 통해 프로그램을 종료한다는 것을 출력한다.

Ex 3-6

#!/bin/sh

mkdir -p $1

cd $1

for i in `seq 0 4`

do

touch file$i.txt

done

tar -cf $1.tar \*.txt

ls

mkdir -p $1

mv $1.tar $1/

cd $1

tar -xvf $1.tar

exit 0

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Mkdir로 인자로 받은 문자가 디렉터리명인 디렉터리를 생성한다. Cd로 해당 디렉터리로 들어간다. for문과 touch를 통해 파일을 반복 생성한다. Tar -cf로 해당 디렉터리 내의 모든 txt파일을 압축하고 압축파일 이름을 인자로 받은 문자로 한다.. Mkdir을 통해 다시 인자로 받은 문자가 디렉터리명인 디렉터리를 생성한다. 압축파일을 새로 생성한 폴더로 옮기고 해당 디렉터리로 들어간다. Tar -xvf를 통해 압축을 풀고 그 과정을 보여준다.

Ex3-7

#!/bin/sh

mkdir -p $1

cd $1

for i in `seq 0 4`

do

file\_name=file$i

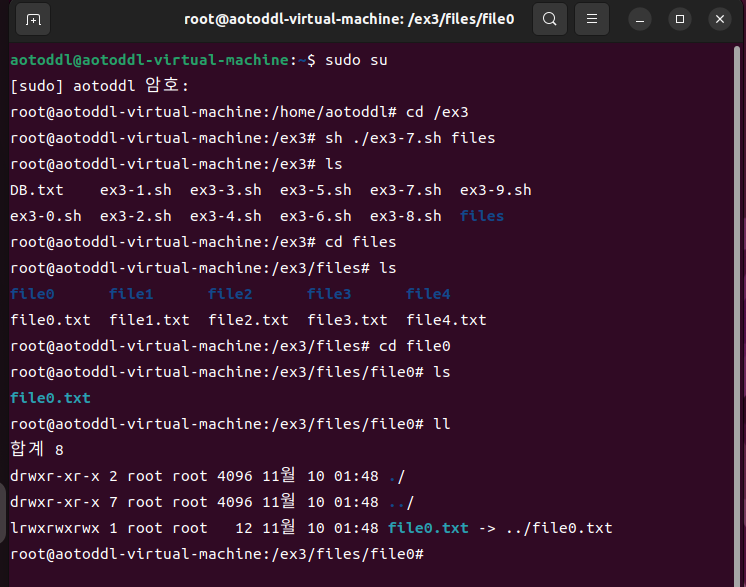
touch $file\_name.txt

mkdir -p $file\_name

ln -s ../$file\_name.txt $file\_name/

done

exit 0



Mkdir로 인자로 받은 문자를 디렉터리명으로 해서 디렉터리를 생성한다. for문과 touch mkdir을 이용해서 파일과 폴더를 생성한다. Ln -s를 이용해서 링크를 생성한다.

Ex3-8.

name=$1

phone=$2

db="DB.txt"

if [ ! -f "$db" ];

then

touch "$db"

fi

echo "$name $phone" >> "$db"

exit 0

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

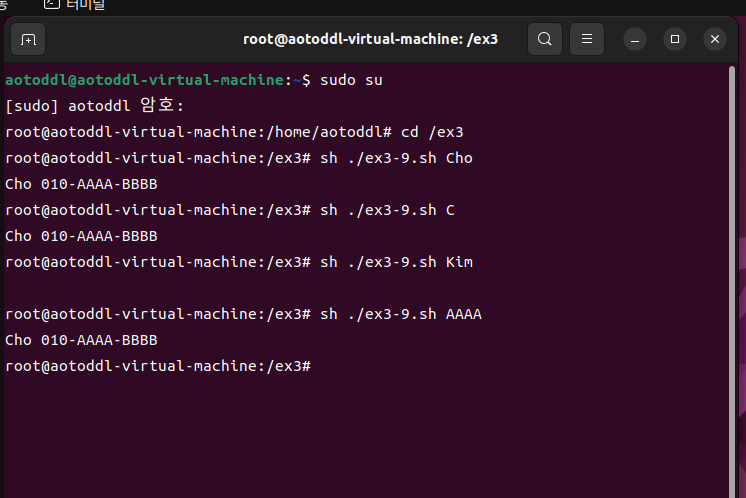
이름과 폰 번호를 인자로 받은 후 DB.txt를 db라는 변수로 두고, if문에서 ! -f를 이용해 파일이 존재하지 않을 때 DB.txt를 touch를 이용하여 생성한다. 그리고 echo와 >>를 이용하여 인자들을 DB.txt에 넣는다.

Ex3-9

#!/bin/sh

echo $(grep $1 DB.txt)

exit 0



Grep을 이용해서 DB.txt에 입력한 인자가 포함되어 있는 줄을 전부 가져온다