System Programming & OS 실습
5. Shell Script

최민국
Dankook University
mgchoi@dankook.ac.kr

Shell Script

- 셸 스크립트.
  - Unix, Linux 등에서 사용하는 일반적인 명령어나 if나 for와 같은 프로그래밍적인 요소로 이루어진 인터프린터 기반의 스크립트 언어

Shell Script

1. 반복 작업 자동화
   - 동일한 작업을 반복 입력할 필요 없이 자동 실행할 수 있다.
   - 예시: 매일 데이터 백업을 자동으로 실행하는 스크립트
     $ tar -czf backup.tar.gz /home/user/data

2. 시스템 관리 및 서버 운영
   - 서버 상태 점검, 로그 정리 등 유지보수 작업을 쉽게 수행할 수 있다.
   - 예시: 현재 CPU 사용량을 확인하는 스크립트
     $ top -b -n1 | grep "Cpu(s)"

3. 여러 명령어 조합 실행
   - 여러 개의 명령어를 하나의 스크립트로 묶어 실행할 수 있다.
   - 예시: /tmp 폴더를 정리하는 스크립트
     $ rm -rf /tmp/*

4. 배치 작업 및 크론 작업
   - 특정 시간에 자동 실행되도록 예약할 수 있다.
   - 예시: 매일 밤 12시에 로그 파일을 정리하는 크론 작업
     $ 0 0 * * * /home/user/clean_logs.sh

5. 개발 및 배포 자동화 (CI/CD)
   - 코드 빌드, 테스트, 배포를 자동화할 수 있다.
   - 예시: Git에서 최신 코드를 가져와 서비스를 재시작하는 배포 스크립트
     $ git pull origin main && systemctl restart myapp

newscript.sh

```
1  #!/bin/bash
2
3  function greeting() {
4    hello="Hello, $name"
5    echo "$hello"
6  }
7
8  echo "Enter name"
9  read name
10
11 val=$(greeting)
12 echo "Return value of the function is $val"
```

Shell Script

- 기초 문법
  - 쉘 스크립트는 맨 윗 줄에 #!/bin/bash를 붙인다.
    - #!: 쉘 인터프리터 지정자 (Shebang)
    - /bin/bash: Bash 쉘이 설치된 경로

  - echo : 문자열을 컴퓨터 터미널에 출력 명령어

  - vim myshell.sh

```
#!/bin/bash
echo “hello, world”
```

  - chmod +x myshell.sh
  - ./myshell.sh

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

echo "hello, world"
choi_gunhee@gunhee-linux-94:~/test$ chmod +x myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ ls
myshell.sh  test2.txt  test.c
choi_gunhee@gunhee-linux-94:~/test$

choi_gunhee@gunhee-linux-94:~/test$ ./myshell.sh
hello, world
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 변수 사용
  - 대소문자 구별
    - VAR과 var은 서로 다른 변수로 인식됨.
  - 숫자로 시작 불가
    - 숫자를 포함할 수 있지만, 변수 이름의 첫 글자는 숫자가 될 수 없음.
  - 모든 값은 문자열로 저장
    - 자료형 없이 문자열처럼 저장되지만, 숫자 연산도 가능함.
  - 일반적으로 자료형을 명시하지 않음
    - int, char 같은 선언 없이 값만 할당하면 됨.

- 변수 사용 시 $ 필요
  변수 값을 참조할 때는 $변수명 또는 ${변수명}을 사용함.

```
echo ${data}
```

- 값을 대입할 때 $ 사용하지 않음
  변수에 값을 할당할 때는 $ 없이 사용.

```
data=mac
```

- = 앞뒤에 공백 없음
  변수 할당 시 = 앞뒤에 공백이 있으면 오류 발생

```
data="abcd"   # 올바른 방식
data = "abcd"  # 오류 발생
```

Shell Script

- 변수 사용
  - 변수 선언 : language = “Korean”
  - 변수 사용 : $Korean

```
#!/bin/bash
language=“Korean”
echo “hello, $language”
```

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash
language="Korean"
echo “hello, $language”

choi_gunhee@gunhee-linux-94:~/test$ ./myshell.sh
“hello, Korean”
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 변수 사용
  - 디렉토리 생성 예제

```
choi_gunhee@gunhee-linux-94:~/test$ ls
myshell.sh  test2.txt  test.c
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

name="kor1 kor2"

mkdir $name
choi_gunhee@gunhee-linux-94:~/test$ ./myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ ls
kor1  kor2  myshell.sh  test2.txt  test.c
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 함수
  - function 키워드 이용

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

function print() {
        echo "hello, function"
        echo $1
}

print "hello, $1 print"
choi_gunhee@gunhee-linux-94:~/test$ ./myshell.sh
hello, function
hello, print
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 전역 변수
  - 스크립트 전체에서 접근 가능한 변수.
  - 특정 함수 또는 코드 블록 밖에서 선언하면, 스크립트 어디에서든 사용 가능.
  - 기본적으로 쉘 스크립트에서 선언된 변수는 자동으로 전역변수가 됨.

- 지역 변수
  - 특정 함수 또는 코드 블록 내에서만 사용 가능.
  - **local** 키워드를 사용하여 선언하면 해당 함수 내에서만 접근 가능.
  - 함수 밖에서는 존재하지 않는 변수로 간주됨.

```
choi_gunhee@gunhee-linux-94:~/test$ vim myshell.sh
choi_gunhee@gunhee-linux-94:~/test$ cat myshell.sh
#!/bin/bash

str1="Hello_str1"

function print() {
        local str2="I am str2"
        echo $str1
        echo $str2
}

print
echo "global $str1"
echo "local $str2"
choi_gunhee@gunhee-linux-94:~/test$ ./myshell.sh
Hello_str1
I am str2
global Hello_str1
local
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 환경 변수
  - 운영체제(특히 리눅스/유닉스)에서 시스템과 사용자 환경을 설정하는 전역변수

환경변수	설명	예시
$HOME	현재 사용자의 홈 디렉토리	/home/minguk
$USER	현재 로그인한 사용자	minguk
$PWD	현재 작업 디렉토리	/home/minguk/docs
$SHELL	기본 쉘 종류	/bin/bash
$PATH	실행 파일 검색 경로	/usr/bin:/bin:/usr/sbin
$LANG	시스템 언어 설정	en_US.UTF-8
$LOGNAME	현재 로그인한 사용자 이름	minguk
$EDITOR	기본 텍스트 편집기	vim
$PS1	프롬프트 형식	[\u@\h \W]\$

Shell Script

예약 변수, 환경 변수
시스템에서 미리 정해둔 변수들이 존재

HOME : 사용자의 홈 디렉토리

PATH : 실행 파일을 찾을 디렉토리 경로

PWD : 현재 작업 중인 디렉토리 경로

USER : 사용자 이름

OSTYPE : 운영체제 종류

```
choi_gunhee@gunhee-linux-94:~/test$ cat system_v.sh
#!/bin/bash

echo "Home : $HOME"
echo "Path : $PATH"
echo "pwd : $PWD"
echo "user : $USER"
echo "OS type : $OSTYPE"
choi_gunhee@gunhee-linux-94:~/test$ ./system_v.sh
Home : /home/choi_gunhee
Path : /home/choi_gunhee/anaconda3/bin:/home/choi_gunhee/anaconda3/condabin:/home/choi_gunhee/anac
onda3/condabin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
games:/snap/bin
pwd : /home/choi_gunhee/test
user : choi_gunhee
OS type : linux-gnu
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 매개 변수
  - $1, $2, $3 … ${10}

```
choi_gunhee@gunhee-linux-94:~/test$ ls
kor1  kor2  myshell.sh  system_v.sh  test2.txt  test.c
choi_gunhee@gunhee-linux-94:~/test$ cat system_v.sh
#!/bin/bash

echo $1
echo $2

choi_gunhee@gunhee-linux-94:~/test$ ./system_v.sh kor1 kor2
kor1
kor2
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- ${변수:=문자열}
  - 변수가 설정되어있지 않거나 NULL이라면 문자열로 치환

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_v.sh
#!/bin/bash

var1="kor1"
var2=""

echo "print var1 : $var1"
echo "print var2 : $var2"

echo ""
echo "print var1 : ${var1:=test2}"
echo "print var2 : ${var2:=test2}"
choi_gunhee@gunhee-linux-94:~/test$ ./str_v.sh
print var1 : kor1
print var2 :

print var1 : kor1
print var2 : test2
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- ${변수:?에러 메세지}
  - 변수가 설정되어있지 않거나 NULL이라면 에러 메세지 출력 후 종료

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_v.sh
#!/bin/bash

str1="str1"
str2=""
error_msg="Sorry"

echo ${str1:?$error_msg}
echo ${str2:?$error_msg}
choi_gunhee@gunhee-linux-94:~/test$ ./str_v.sh
str1
./str_v.sh: line 8: str2: Sorry
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- ${변수:=문자열}
  - 그 외 변환

변환	설명
${변수-문자열}	변수가 설정되지 않은 경우, 문자열을 변수로 치환
${변수:-문자열}	변수가 설정되지 않았거나 NULL인 경우 문자열을 변수로 치환
${변수=문자열}	변수가 설정되지 않은 경우, 변수에 문자열을 저장하고 치환
${변수:=문자열}	분수가 설정되지 않았거나 NULL인 경우 문자열을 변수에 저장하고 치환
${변수:?에러 메세지}	변수가 설정되지 않았거나 NULL인 경우, 에러메세지 출력하고 종료
${변수:시작위치}	변수값이 문자열인 경우, 시작 위치부터 문자열 길이 끝까지 출력
${변수:시작위치:길이}	변수값이 문자열인 경우, 시작 위치부터 길이까지 출력

Shell Script

- 문자열 패턴, 변경

패턴, 변경	설명
${변수#패턴}	변수 **앞**에서부터 **처음** 찾은 패턴과 일치하는 패턴 **앞의 모든** 문자열 제거
${변수##패턴}	변수 **앞**에서부터 **마지막** 찾은 패턴과 일치하는 패턴 **앞의 모든** 문자열 제거
${변수%패턴}	변수 **뒤**에서부터 **처음** 찾은 패턴과 일치하는 패턴 **뒤의 모든** 문자열 제거
${변수%%패턴}	변수 **뒤**에서부터 **마지막** 찾은 패턴과 일치하는 패턴 **뒤의 모든** 문자열 제거
${#변수}	변수의 길이
${변수/찾는문자열/바꿀문자열}	변수에서 찾는 문자열을 바꿀 문자열로 변경, 없으면 삭제
${변수/#찾을문자열/바꿀문자열}	변수에서 문자열에서 찾을 문자열 **시작과** 맞으면 문자열 변경
${변수/%찾을문자열/바꿀문자열}	변수에서 문자열에서 찾을 문자열 **마지막과** 맞으면 문자열 변경

Shell Script

- 문자열 패턴, 변경

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_change.sh
#!/bin/bash

str1="/etc/test1/hello.txt"

echo $str1
echo ${str1%/*}
echo ${str1##*/}
echo ${#str1}
choi_gunhee@gunhee-linux-94:~/test$ ./str_change.sh
/etc/test1/hello.txt
/etc/test1
hello.txt
20
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 문자열 패턴, 변경

```
choi_gunhee@gunhee-linux-94:~/test$ cat str_change.sh
#!/bin/bash

str1="hello choi gunhee hello choi"

echo ${str1/hello/hi}
echo ${str1//hello/hi}
echo ${str1/hello}
echo ${str1//hello}
echo ${str1/#he/what?}
echo ${str1/%lo/what??}
choi_gunhee@gunhee-linux-94:~/test$ ./str_change.sh
hi choi gunhee hello choi
hi choi gunhee hi choi
choi gunhee hello choi
choi gunhee choi
what?llo choi gunhee hello choi
hello choi gunhee hello choi
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 조건문

  If [첫 번째 조건식]
  then
  　　수행문
  elif [두 번째 조건식]
  then
  　　수행문
  else
  　　수행문
  fi

```
choi_gunhee@gunhee-linux-94:~/test$ cat if_case.sh
#!/bin/bash
v1=10
v2=10

if [ $v1 = $v2 ]
then
      echo True
else
      echo False
fi
choi_gunhee@gunhee-linux-94:~/test$ ./if_case.sh
True
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 조건문

  case $변수 in
  조건값 1)
  수행문 ;;
  조건값 2)
  수행문 ;;
  *)
  수행문
  esac

```
choi_gunhee@gunhee-linux-94:~/test$ vim if_case.sh
choi_gunhee@gunhee-linux-94:~/test$ cat if_case.sh
#!/bin/bash

case $1 in
        hi)
        echo hi
        ;;
        hello)
        echo hello
        ;;
        *)
esac
choi_gunhee@gunhee-linux-94:~/test$ ./if_case.sh hi
hi
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 반복문

  for 변수 in [범위]
  do
  　　반복할 수행문
  done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

for num in 1 2 3 4 5
do
        echo $num;
done
choi_gunhee@gunhee-linux-94:~/test$ ./iter.sh
1
2
3
4
5
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 반복문
  for 변수 in [범위]
  do
  　　반복할 수행문
  done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

for file in $HOME/*
do
      echo $file;
done
choi_gunhee@gunhee-linux-94:~/test$ ./iter.sh
/home/choi_gunhee/anaconda3
/home/choi_gunhee/Anaconda3-2021.11-Linux-x86_64.sh
/home/choi_gunhee/cache-trace
/home/choi_gunhee/cscope.files
/home/choi_gunhee/cscope.out
/home/choi_gunhee/data
/home/choi_gunhee/Desktop
/home/choi_gunhee/Documents
/home/choi_gunhee/Downloads
/home/choi_gunhee/Easy-Shell-Script
```

Shell Script

- 반복문
  for 변수 in [범위]
  do
  　　반복할 수행문
  done

```
choi_gunhee@gunhee-linux-94:~/test$ cat ./iter.sh
#!/bin/bash

for num in {1..10..2}
do
        echo $num;
done
choi_gunhee@gunhee-linux-94:~/test$ ./iter.sh
1
3
5
7
9
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 반복문
  for 변수 in [범위]
  do
  　　반복할 수행문
  done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

arr=("a" "b" "c" "d" "e")

for str in ${arr[@]}
do
        echo $str;
done
choi_gunhee@gunhee-linux-94:~/test$ ./iter.sh
a
b
c
d
e
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 반복문

  while [$변수1 연산자 $변수2]
  do
  　　수행문
  done

```
choi_gunhee@gunhee-linux-94:~/test$ cat iter.sh
#!/bin/bash

n=1

while [ $n -lt 5 ]
do
        echo $n
        n=$((n+1))
done
choi_gunhee@gunhee-linux-94:~/test$ ./iter.sh
1
2
3
4
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 연산자

연산자	설명
if [-z $변수]	문자열 길이가 0이면 참
if [-n $변수]	문자열 길이가 0아니면 참
if [$변수1 -eq $변수2]	변수1과 변수2의 값이 같으면 참
if [$변수1 -ne $변수2]	변수1과 변수2의 값이 다르면 참
if [$변수1 -gt $변수2]	변수1이 변수2보다 크면 참
if [$변수1 -ge $변수2]	변수1일 변수2보다 크거나 같으면 참
if [$변수1 -lt $변수2]	변수1이 변수2보다 작으면 참
if [$변수1 -le $변수2]	변수1이 변수2보다 작거나 같으면 참
>, >=, <, <=	사용시, if (($변수 1 > $변수2))처럼, ()로 한번 더 감싸야한다.

Shell Script

- grep
  - 특정 디렉토리, 로그 등에서 특정 문자열을 찾는 명령어
  - grep [옵션] 패턴 파일

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee       4096  5월 18 17:42 anaconda3
-rw-rw-r--  1 choi_gunhee choi_gunhee  608680744 11월 18  2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x  5 choi_gunhee choi_gunhee       4096  8월 19 11:19 cache-trace
-rw-rw-r--  1 choi_gunhee choi_gunhee   18930510  3월 24 00:36 cscope.files
-rw-rw-r--  1 choi_gunhee choi_gunhee 2667770462  3월 24 00:37 cscope.out
drwxrwxr-x  2 choi_gunhee choi_gunhee       4096  8월 19 22:57 data
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Desktop
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Documents
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee       4096  9월 12 19:04 Easy-Shell-Script
drwxrwxr-x  3 choi_gunhee choi_gunhee       4096  5월 18 17:40 kangaroo_project
drwxrwxr-x  7 choi_gunhee choi_gunhee       4096  8월 24 15:39 memcached-1.6.16
-rw-rw-r--  1 choi_gunhee choi_gunhee    1054877  8월  4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x  8 choi_gunhee choi_gunhee       4096  6월 17 10:16 MyProject
-rw-------  1 choi_gunhee choi_gunhee        989  6월  7 15:41 nohup.out
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Pictures
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Public
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  3월  8  2022 snap
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Templates
drwxrwxr-x  4 choi_gunhee choi_gunhee       4096  9월 12 21:49 test
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  6월  7 19:32 u3_ctrl
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Videos
choi_gunhee@gunhee-linux-94:~/test$ grep -i cscope list.txt
-rw-rw-r--  1 choi_gunhee choi_gunhee   18930510  3월 24 00:36 cscope.files
-rw-rw-r--  1 choi_gunhee choi_gunhee 2667770462  3월 24 00:37 cscope.out
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- grep
  - 특정 디렉토리, 로그 등에서 특정 문자열을 찾는 명령어
  - grep [옵션] 패턴 파일

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee       4096  5월 18 17:42 anaconda3
-rw-rw-r--  1 choi_gunhee choi_gunhee  608680744 11월 18  2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x  5 choi_gunhee choi_gunhee       4096  8월 19 11:19 cache-trace
-rw-rw-r--  1 choi_gunhee choi_gunhee   18930510  3월 24 00:36 cscope.files
-rw-rw-r--  1 choi_gunhee choi_gunhee 2667770462  3월 24 00:37 cscope.out
drwxrwxr-x  2 choi_gunhee choi_gunhee       4096  8월 19 22:57 data
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Desktop
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Documents
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee       4096  9월 12 19:04 Easy-Shell-Script
drwxrwxr-x  3 choi_gunhee choi_gunhee       4096  5월 18 17:40 kangaroo_project
drwxrwxr-x  7 choi_gunhee choi_gunhee       4096  8월 24 15:39 memcached-1.6.16
-rw-rw-r--  1 choi_gunhee choi_gunhee    1054877  8월  4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x  8 choi_gunhee choi_gunhee       4096  6월 17 10:16 MyProject
-rw-------  1 choi_gunhee choi_gunhee        989  6월  7 15:41 nohup.out
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Pictures
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Public
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  3월  8  2022 snap
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Templates
drwxrwxr-x  4 choi_gunhee choi_gunhee       4096  9월 12 21:49 test
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  6월  7 19:32 u3_ctrl
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월  8  2022 Videos
choi_gunhee@gunhee-linux-94:~/test$ grep -i cscope list.txt
-rw-rw-r--  1 choi_gunhee choi_gunhee   18930510  3월 24 00:36 cscope.files
-rw-rw-r--  1 choi_gunhee choi_gunhee 2667770462  3월 24 00:37 cscope.out
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- find
  - 파일을 찾아주는 명령어
  - find [옵션] [경로] [표현식]

```
choi_gunhee@gunhee-linux-94:~/test$ ls
if_case.sh  kor1  list.txt  msg          str_change.sh  system_v.sh  test.c
iter.sh     kor2  log.sh    myshell.sh   str_v.sh       test2.txt
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list.txt
./list.txt
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list
choi_gunhee@gunhee-linux-94:~/test$ find ./ -name list*
./list.txt
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- awk
  - 특정 인덱스 문자열 출력
  - awk [옵션] 패턴(액션) [대상파일]

```
choi_gunhee@gunhee-linux-94:~/test$ cat list.txt
total 3219264
drwxrwxr-x 28 choi_gunhee choi_gunhee       4096  5월  18 17:42 anaconda3
-rw-rw-r--  1 choi_gunhee choi_gunhee  608680744 11월  18  2021 Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x  5 choi_gunhee choi_gunhee       4096  8월  19 11:19 cache-trace
-rw-rw-r--  1 choi_gunhee choi_gunhee   18930510  3월  24 00:36 cscope.files
-rw-rw-r--  1 choi_gunhee choi_gunhee 2667770462  3월  24 00:37 cscope.out
drwxrwxr-x  2 choi_gunhee choi_gunhee       4096  8월  19 22:57 data
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Desktop
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Documents
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Downloads
drwxrwxr-x 13 choi_gunhee choi_gunhee       4096  9월  12 19:04 Easy-Shell-Script
drwxrwxr-x  3 choi_gunhee choi_gunhee       4096  5월  18 17:40 kangaroo_project
drwxrwxr-x  7 choi_gunhee choi_gunhee       4096  8월  24 15:39 memcached-1.6.16
-rw-rw-r--  1 choi_gunhee choi_gunhee    1054877  8월   4 12:36 memcached-1.6.16.tar.gz
drwxrwxr-x  8 choi_gunhee choi_gunhee       4096  6월  17 10:16 MyProject
-rw-------  1 choi_gunhee choi_gunhee        989  6월   7 15:41 nohup.out
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Pictures
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Public
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  3월   8  2022 snap
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Templates
drwxrwxr-x  4 choi_gunhee choi_gunhee       4096  9월  12 21:49 test
drwxr-xr-x  3 choi_gunhee choi_gunhee       4096  6월   7 19:32 u3_ctrl
drwxr-xr-x  2 choi_gunhee choi_gunhee       4096  3월   8  2022 Videos
```

```
choi_gunhee@gunhee-linux-94:~/test$ awk '{print $1, $9'} list.txt
total
drwxrwxr-x anaconda3
-rw-rw-r-- Anaconda3-2021.11-Linux-x86_64.sh
drwxrwxr-x cache-trace
-rw-rw-r-- cscope.files
-rw-rw-r-- cscope.out
drwxrwxr-x data
drwxr-xr-x Desktop
drwxr-xr-x Documents
drwxr-xr-x Downloads
drwxrwxr-x Easy-Shell-Script
drwxrwxr-x kangaroo_project
drwxrwxr-x memcached-1.6.16
-rw-rw-r-- memcached-1.6.16.tar.gz
drwxrwxr-x MyProject
-rw------- nohup.out
drwxr-xr-x Pictures
drwxr-xr-x Public
drwxr-xr-x snap
drwxr-xr-x Templates
drwxrwxr-x test
drwxr-xr-x u3_ctrl
drwxr-xr-x Videos
choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- User 계정 만들기

```
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd | grep choi_gunhee | wc -l
1
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd | grep choi_gunhee
choi_gunhee:x:1000:1000:choi_gunhee,,,:/home/choi_gunhee:/bin/bash
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

Shell Script

- User 계정 만들기
  -

```
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
1
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
choi_gunhee:x:1000:1000:choi_gunhee,,,:/home/choi_gu
choi_gunhee@gunhee-linux-94:~/test$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

```
choi_gunhee@gunhee-linux-94:~/test$ cat addUser_script.sh
#!/bin/bash

if [[ -n $1 ]] && [[ -n $2 ]]
then
        UserId=$1
        Password=$2

        if [[ $(cat /etc/passwd | grep $UserId | wc -l) == 0 ]]
        then
          useradd -m $UserId
          echo $Password | passwd $UserId --stdin
        else
          echo "this user is existing : $UserId"
        fi
else
        echo -e "Please input user id and password"
fi

choi_gunhee@gunhee-linux-94:~/test$
```

Shell Script

- 사용자별 사용량 체크
  -

```
choi_gunhee@gunhee-linux-94:~/test$ cat ./userUsage.sh
#!/bin/bash

dir=/home

for userPath in $dir/*
do
        user=${userPath##*/}
        echo $user
        size=$(ls -l /home/$user | grep $user | awk '{ sum += $5 } END { print sum }')
        if [ -z $size ]
        then
                echo $user size is 0
        else
                echo $user size is $size
        fi
        echo ""
done


choi_gunhee@gunhee-linux-94:~/test$
```