Shell Programming Exercises

1. 디렉토리 생성 프로그램

1. 목적

반복문을 사용하여 복수의 하위 디렉토리를 생성

2. 상세

- 디렉토리 내에 다시 하위디렉토리 생성시 중첩 반복문 사용

3. 명령어 정리

- for 반복문 사용

```
for [변수] in [List]
do
    command
done
```

- 디렉토리 생성

```
mkdir test
```

1. 디렉토리 생성 프로그램

```
-bash-3.2# ./dir.sh
dir1 Created...
        subdir1 Created...
        subdir2 Created...
        subdir3 Created...
        subdir4 Created...
dir2 Created...
        subdir1 Created...
        subdir2 Created...
        subdir3 Created...
        subdir4 Created...
dir3 Created...
        subdir1 Created...
        subdir2 Created...
        subdir3 Created...
        subdir4 Created...
dir4 Created...
        subdir1 Created...
        subdir2 Created...
        subdir3 Created...
        subdir4 Created...
```

2. 간단한 계산기 프로그램

1. 목적

계산기 기능을 수행할 수 있는 프로그램을 작성

2. 상세

- 간단한 사칙연산을 수행할 수 있는 프로그램
- 변수를 입력 받는 방법 / 위치매개변수를 사용하는 방법

3. 명령어 정리

변수 입력

```
read [변수명]
```

조건문

```
if [ 변수 = 값 ]
then
        command
fi
```

case

수식 계산

```
expr [변수1] + [변수2]
```

case 선택문

```
case [변수] in
        값1) command ;;
        값2) command ;;
        *) command ;;
esac
```

2. 간단한 계산기 프로그램 (1)

```
-bash-3.2# ./calc.sh
Enter A : 10
Enter B : 20
Select operator 1)+ 2)- 3)* 4)/
Select one : 1
10 + 20 = 30
-bash-3.2# ./calc.sh
Enter A : 20
Enter B : 30
Select operator 1)+ 2)- 3)* 4)/
Select one : 3
20 * 30 = 600
-bash-3.2# ./calc.sh
Enter A : 10
Enter B : 3
Select operator 1)+ 2)- 3)* 4)/
Select one : 4
10 / 3 = 3.333
```

2. 간단한 계산기 프로그램 (2)

```
-bash-3.2# ./calc2.sh 10 + 3
10 + 3 = 13
-bash-3.2# ./calc2.sh 10 - 3
10 - 3 = 7
-bash-3.2# ./calc2.sh 10 '*' 3
10 * 3 = 30
-bash-3.2# ./calc2.sh 10 / 3
10 / 3 = 3.333
```

3. 사용자 추가 프로그램

1. 목적

복수의 사용자를 추가할 수 있는 프로그램

2. 상세

- useradd 명령어를 사용하는 방법
- /etc/passwd , /etc/shadow 파일을 수정하는 방법 (OS에 따라 적용가능)

3. 명령어 정리

반복문

```
while [ 조건문 ]
do
        command
done
```

사용자 추가

```
useradd –m –d /export/home/[사용자ID] -s /bin/bash [사용자ID]
```

3. 사용자 추가 프로그램

```
bash-3.2# ./adduser.sh
Enter Your Username? [user] : gj
User basename : gj
Is correct ? [y/n]: y
gj2000 and password created...
gj2001 and password created...
gj2002 and password created...
-bash-3.2# tail -3 /etc/passwd
gj2000:x:10000:10::/export/home/gj2000:/bin/bash
gj2001:x:10001:10::/export/home/gj2001:/bin/bash
gj2002:x:10002:10::/export/home/gj2002:/bin/bash
-bash-3.2# tail -3 /etc/shadow
gj2000:dTMUafFG4/SOQ::::::
gj2001:dTMUafFG4/SOQ::::::
gj2002:dTMUafFG4/SOQ::::::
```

4. 주요 파일 점검 프로그램

1. 목적

중요한 파일을 확인하는 스크립트

2. 상세

- 파일을 복사하여 사본을 보관
- 원본 파일이 없어졌을 경우 사본을 원본 위치로 복사
- diff를 사용하여 파일의 변경내용을 확인
- 변경사항 발생시 알림

3. 명령어 정리

파일 비교 : **diff [파일1] [파일2]**

파일 복사 : **cp [원본파일] [사본파일]**

4. 주요 파일 점검 프로그램

```
-bash-3.2# cat > test.txt
qwer
-bash-3.2# ./filechk.sh
Backup file Created...
-bash-3.2# ./filechk.sh
Backup File exist...
file check complete!
-bash-3.2# cat >> test.txt
asdf
-bash-3.2# ./filechk.sh
Backup File exist...
file has changed!
-bash-3.2# rm test.txt
-bash-3.2# ./filechk.sh
Backup File exist...
File has removed!
Restore Backup...
-bash-3.2# cat test.txt
qwer
```

5. 시스템 상태 체크 프로그램

1. 목적

주기적으로 시스템 상태를 확인하여 파일에 저장하는 프로그램

2. 상세

- 주기적으로 시스템 상태 모니터링 프로그램을 수행
- 결과를 실행 시간에 따른 파일이름으로 저장
- 사용자가 중지할 때 까지 반복하도록 실행

3. 명령어 정리

파일 생성 : **touch [파일이름]**

현재 년월일시분초 문자열 생성 : **date '+%y%m%d%H%M%S'**

메모리 상태 확인 : **vmstat**

프로세스 정보 확인 : **ps –ef**

디스크 상태 확인 : **df -h**

5. 시스템 상태 체크 프로그램

```
-bash-3.2# ./systemchk.sh
LOG_160331193610.log has been created...
LOG_160331193620.log has been created...
-bash-3.2# head LOG_160331193620.log
LOGGING TIME ====================
2016년 3월 31일 목요일 오후 07시 36분 20초

MEMORY STATUS ==================
 kthr      memory            page            disk        faults      cpu
 r b w   swap  free  re  mf pi po fr de sr f0 s0 s1 --   in   sy   cs us sy id
 0 0 16 1199620 851692 3 14  3  1  2  0 48 -0  1 -0  0  155  314  243  0 53 47

DISK STATUS ====================
파일시스템          크기   사용   가용   용량    설치지점
```

6. 파일 이름 일괄 변경 프로그램

1. 목적

현재 디렉토리에서 지정된 파일들의 확장자를 일괄 변경

2. 상세

- 동일한 확장자를 가진 파일을 확인
- 확장자 일괄 변경 수행

3. 명령어 정리

파일 이름 변경 : **mv [예전파일명] [새로운파일명]**

변수 뒷자리에서 특정 패턴 제거

ex) **name = test.txt**

echo $name → **test.txt**

echo ${name%*txt} → **test.**

echo ${name%*.txt} → **test**

6. 파일 이름 일괄 변경 프로그램

```
-bash-3.2# touch a.txt b.txt c.txt
-bash-3.2# ls *.txt
a.txt     b.txt     c.txt     test.txt
-bash-3.2# ls *.bak
*.bak: 해당 파일이나 디렉토리가 없음
-bash-3.2# ./namechanger.sh
Extention from : txt
Extension to : bak
a.txt found...
changed to a.bak
b.txt found...
changed to b.bak
c.txt found...
changed to c.bak
-bash-3.2# ls *.txt
*.txt: 해당 파일이나 디렉토리가 없음
-bash-3.2# ls *.bak
a.bak        b.bak        c.bak
```

7. 디스크 상태 실시간 모니터링 프로그램

1. 목적

디스크 상태를 실시간으로 모니터링 하는 프로그램

2. 상세

- 디스크 상태를 지속적으로 모니터링
- 기본 명령어 형식이 아닌 지정된 형식으로 출력
- /export/home (사용자 홈 디렉토리) 위치를 모니터링
- 사용량이 일정 %를 초과하면 경고

3. 명령어 정리

디스크 상태 확인 : **df –h**

임의크기 파일 생성 : **dd if=/dev/zero of=/export/home/100m bs=1000000 count=1**

awk '/패턴/ BEGIN{ } { } END{ }' 대상파일

명령어 | awk '/패턴/ BEGIN{ } { } END{ }'

7. 디스크 상태 실시간 모니터링 프로그램

```
-bash-3.2# ./diskchk.sh
Disk Size : 6.2G
Used : 6.5M
Available : 6.1G
Disk Size : 6.2G
Used : 6.5M
Available : 6.1G
-bash-3.2# dd if=/dev/zero of=/export/home/100m bs=100000000 count=1
1+0 레코드 입력
1+0 레코드 출력
-bash-3.2# ./diskchk.sh
Disk Size : 6.2G
Used : 102M
Available : 6.0G
>>>>> Disk Usage Percentage is now 2%! <<<<<
```

8. 휴지통 만들기 프로그램

1. 목적

rm 명령어 사용시 즉시 삭제되므로, 휴지통 기능을 하는 명령어 구형

2. 상세

- 삭제명령을 내린 파일은 /recycle 로 이동
- 첫 번째 위치매개변수로 –r을 입력할 경우 파일 복원
- 첫 번째 위치매개변수로 –l 을 입력할 경우 휴지통 파일 리스트 출력
- 파일 복원 시 원래 위치로 복원

3. 명령어 정리

파일 삭제 : rm [파일명]

8. 휴지통 만들기 프로그램

```
-bash-3.2# touch fileA
-bash-3.2# ls -l fileA
-rw-r--r--  1 root    root        0  3월 31일  19:42 fileA
-bash-3.2# ./myrm.sh fileA
Recycle Bin : /recycle
fileA has moved to Recycle Bin...
-bash-3.2# ls -l fileA
fileA: 해당 파일이나 디렉토리가 없음
-bash-3.2# ls -l /recycle
총 2
-rw-r--r--  1 root    root        0  3월 31일  19:42 fileA
-rw-r--r--  1 root    root        5  3월 31일  19:42 fileA_path
bash-3.2# ./myrm.sh -l
Recycle Bin : /recycle
fileA
-bash-3.2# ./myrm.sh -r fileA
Recycle Bin : /recycle
 file has beed restored...
-bash-3.2# ls -l /recycle
총 0
-bash-3.2# ls -l fileA
-rw-r--r--  1 root    root        0  3월 31일  19:42 fileA
```

9. 실행중인 프로세스 중지 프로그램

1. 목적

프로세스의 이름과 시그널 번호를 통해 프로세스를 종료

2. 상세

- 프로세스의 이름으로 PID를 조회
- 복수 프로세스 종료 가능
- KILL, TERMINATE 시그널을 선택하여 시그널 전송

3. 명령어 정리

ps –e : 전체 프로세스 조회

kill -9 : KILL 시그널 전송 (강제종료, 무시불가)

kill -15 : TERM 시그널 전송 (강제성 없음)

9. 실행중인 프로세스 중지 프로그램

```
-bash-3.2# ps -ef | grep sleep
   root 11387  5540   0 19:44:54 pts/4      0:00 sleep 1000
   root 11386  5540   0 19:44:53 pts/4      0:00 sleep 1000
   root 11388  5540   0 19:44:54 pts/4      0:00 sleep 1000
-bash-3.2# ./killproc.sh sleep
Process no : 11387
Process Name: sleep
Select Kill/Term/Cancel (K/T/C) k
Kill Signal sent...

Process no : 11386
Process Name: sleep
Select Kill/Term/Cancel (K/T/C) t
Term Signal sent...

Process no : 11388
Process Name: sleep
Select Kill/Term/Cancel (K/T/C) c
Canceled...

[1]  종료됨(Terminated)   sleep 1000
[2]- 중단됨(Killed)       sleep 1000
```

◆ Shell Programming 참고할 만한 곳

- Shell 프로그래밍의 기본

https://wiki.kldp.org/Translations//html/Shell_Programming-KLDP

- 고급 Bash 스크립팅 가이드

https://wiki.kldp.org/HOWTO/html/Adv-Bash-Scr-HOWTO/

$ google에서 다음을 검색

site:kldp.org shell프로그래밍