

UNIX시스템 과제 7

1. 다음과 같이 모든 인수의 합을 계산하여 출력하는 shell program을 작성하시오.

```
$ add 1 2 3 4
```

```
10
```

2. 키보드로 입력된 파일이름 또는 디렉토리 이름에 대해서 다음과 파일 속성을 판단하여 출력하는 shell program을 작성하시오

```
$ filetype
```

```
파일/디렉토리 이름 입력 [enter]
```

(출력 유형)

...은 존재하지 않습니다.

...은 디렉토리/심볼릭링크/문자장치/블록장치/파이프/소켓/보통파일 입니다.

다음은 각각 입력하여 작성한 shell 프로그램의 동작을 확인하세요.

```
/bin/awk /dev/tty /dev/sda /usr/bin ~gdhong/myfifo /etc/printcap
```

3. 강의 자료의 safe editor(파일이름 svi)를 분석하고 실행시키시오.

4. 다음과 같이 동작하는 track 셸 프로그램을 작성해보자.

```
Script: track [-ncount][-tpause] userId
```

track monitors the specified user's login and logout sessions. Every 'pause' number of seconds, track scans the system and makes a note of who is currently logged on. If the specified user has logged on or logged off since the last scan, this information is displayed to standard output. track operates until 'count' scans have completed. By default, 'pause' is 20 seconds and 'count' is 10,000 scans. track is usually executed in the background with its standard output redirected.

다음은 이러한 동작을 하는 shell program을 작성한 것이다. 이 프로그램을 분석하고 이 프로그램을 작성하여 동작을 확인하시오. (이 프로그램은 track이외에 보조 프로그램인 track.cleanup과 보조 파일인 track.sed가 함께 사용된다.)

track (실행 파일 - 실행허가권 부여)

```
#!/bin/sh
pause=20          # default pause between scans
loopCount=10000  # default scan count
error=0          # error flag
for arg in $*     # parse command line arguments
do
  case $arg in
    -t*)          # time
      pause=`expr substr $arg 3 10` # extract number
      ;;
    -n*)          # scan count
      loopCount=`expr substr $arg 3 10` # extract number
      ;;
    *)
      user=$arg   # user name
      ;;
  esac
done
if [ ! "$user" ] # check a user id was found
then
  error=1
fi
if [ $error -eq 1 ] # display error message if error(s) found
then
  cat << ENDOFERROR # display usage message
usage: track [-n#] [-t#] userId
ENDOFERROR
  exit 1           # terminate shell
fi
trap 'track.cleanup $$; exit $exitCode' 0 # trap on exit
trap 'exitCode=1; exit' 2 3             # trap on INT/QUIT
echo -n > .track.old.$$ #zero the old track file.
count=0
while [ $count -lt $loopCount ]
do
  who | grep $user | sort > .track.new.$$ #scan system
  diff .track.new.$$ .track.old.$$ | sed -f track.sed > .track.report.$$
  if [ -s .track.report.$$ ]             # only report changes
  then                                     # display report
    echo track report for ${user}:
    cat .track.report.$$
  fi
  mv .track.new.$$ .track.old.$$         # remember current state
  sleep $pause                            # wait a while
  count=`expr $count + 1`                 # update scan count
done
exitCode=0                                # set the exit code
```

track.cleanup (실행파일 - 실행허가권 부여)

```
echo stop tracking
rm -f .track.old.$1 .track.new.$1 .track.report.$1
```

track.sed

```
/^[0-9].*/d
/^----/d
s/^</login/
s/^>/logout/
```