

Unix shell scripts

(shell utilities for Unix)

2005년 03월 21일

Sunny Kwak

sunnykwak@hanmail.net

sunnykwak@egloos.com

Copyrights

© 2004 Sunny Kwak. All rights reserved.

Other disclaimers

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Version

Version 1.0 (draft)

History

2004년 11월 23일 화요일 First created.

Table of Contents

1. INTRODUCTION	4
1.1 OBJECTIVES	4
1.2 TARGET AUDIENCE	4
2. SHELL STARTUP SCRIPT	4
2.1 STARTUP SCRIPT의 역할	4
2.2 STARTUP SCRIPT의 위치 및 실행 방식	4
2.2.1 Bourne shell(sh)	4
2.2.2 C shell(csh)	4
2.3 STARTUP SCRIPT EXAMPLE.....	4
3. UTILITY SHELL SCRIPTS.....	6
3.1 KILLP (KILL NAMED PROCESSES)	6
3.2 FG (FIND AND GREP FILES)	8
3.3 PORT_TRAFFIC (COUNT NETWORK PORT TRAFFIC).....	8
3.4 TRACE_MEM (TRACE MEMORY USAGE)	9
3.5 MON_CPU (MONITOR CPU LOAD)	10
3.6 ZIP_N_EXT (ZIP OR EXTRACT FILES).....	11
3.7 RESTART_ALL_WEBSVR (RESTART ALL WEB SERVERS)	12
3.8 ORACLE_START8 (STARTUP ORACLE 8)	13
3.9 ORACLE_STOP8(SHUTDOWN ORACLE 8)	14
3.10 ORACLE_START9 (STARTUP ORACLE 9)	14
3.11 ORACLE_STOP9 (SHUTDOWN ORACLE 9)	14
3.12 RESTART_TOMCAT	15
3.13 MON_LOG (MONITOR LOG).....	15

1. Introduction

1.1 Objectives

“Unix shell scripts” 문서에서 기술하는 각종 쉘 스크립트는 Unix 시스템 상에서 서비스 유지보수, 시스템 상태 점검, 각종 자료 관리 등에 유용한 유틸리티 프로그램들이다. 각 스크립트 프로그램(script program)의 소스에 대한 설명 및, 용도와 사용법, 응용 방안 등에 대해서 설명하고 있다.

1.2 Target Audience

개발자 및 시스템 관리자

2. Shell startup script

2.1 startup script의 역할

Unix startup script는 일반적으로 다음과 같은 역할을 수행한다.

Shell 동작을 위한 각종 변수 선언 및 변수 값의 설정
실행 파일 검색 경로의 설정
터미널 설정
환경 변수 설정
로그인 시 수행되기를 원하는 명령어 수행

2.2 startup script의 위치 및 실행 방식

2.2.1 Bourne shell(sh)

login shell은 홈 디렉토리 상의 .profile을 읽어 환경 설정.
subshell은 .profile을 이용하지 않고 그 parent shell로부터 환경 정보를 상속.

2.2.2 C shell(csh)

모든 shell(subshell 포함)은 홈 디렉토리 상의 .cshrc 파일을 읽어 처리
login shell의 경우, 먼저 홈 디렉토리 상의 .login 파일을 읽어 처리
login shell의 경우, logout 시 홈 디렉토리 상의 .logout 파일을 읽어 처리.

2.3 startup script example

샘플 shell startup 파일은 Solaris OS 및 C shell을 사용하는 환경을 기준으로 작성되었다. 샘플 shell startup 파일에서는 기본적인 환경 설정 및 각종 alias, Oracle 및 Sybase DB 연결을 위한 설정과 Java 개발을 위한 설정이 포함되어 있다.

```
# @(#)cshrc

umask 022

stty erase ^H
```

```
if( $?prompt ) then
    set history=32
endif

set filec
setenv DISPLAY unix:0.0
setenv OPENWINHOME /usr/openwin
setenv EDITOR /usr/ucb/vi
setenv LANG ko

limit coredumpsize 0

#####
# aliases as utility
#####
alias setprompt 'set prompt="[`whoami`@`hostname`:~]$ " '
alias cd 'cd \!*; setprompt'
alias ls '/bin/ls -F'
alias ll '/bin/ls -als | more'
alias df 'df -k'
alias h 'history \!* | more'
alias so 'source ~/.cshrc'
alias pushd 'pushd \!*; setprompt'
alias popd 'popd; setprompt'
alias systemoff 'repeat 9 sync ; init 0'
alias rebooting 'repeat 9 sync ; reboot -- -r'

#####
# aliases for development
#####
alias iisql 'unsetenv LANG; isql -Udevusr -Pdevusr'

#####
# sybase variable
#####
setenv SYBASE /home0/sybase
setenv DSQUERY ZEUS

#####
# oracle parameter
#####
setenv ORACLE_SID ORA734
setenv ORACLE_HOME /home2/oracle/app/oracle/product/7.3.4
setenv ORACLE_TERM sun
setenv ORACLE_OWNER oracle
setenv NLS_LANG AMERICAN_AMERICA.KO16KSC5601
setenv TNS_ADMIN $ORACLE_HOME/network/admin
```

```

set path=(/usr/sbin /usr/ccs/bin /usr/bin /sbin /etc /usr/dt/bin \
/usr/openwin/bin /opt/SUNWsprow/bin \
/usr/local/bin /usr/local/lib /etc /usr/ucb \
/usr/local/etc $ORACLE_HOME/bin \
$SYBASE/bin $SYBASE/OCS-12_5/bin \
$SYBASE/install $SYBASE/lib /usr/java/bin . )

setenv LD_LIBRARY_PATH
usr/dt/lib:/usr/openwin/lib:/usr/lib:/usr/ucblib:/usr/local/lib:$SYBASE/
SE-12_5/lib:$SYBASE/OCS-12_5/lib:$ORACLE_HOME/lib

setenv CLASSPATH
usr/java/lib/tools.jar:/home1/work/iplanet/bin/https/jar/servlet.jar:/ho
me1/work/public/jdbc/jdbc2_0-
stdext.jar:/home1/work/public:/home1/work/public/taglibs

setprompt

```

3. Utility Shell scripts

3.1 killp (kill named processes)

Description

실행 중인 unix 프로세스 중에서 killp script의 인자(arguments)로 지정한 이름을 가진 프로세스를 찾아 kill 명령을 수행한다. 이 때, 종료하고자 하는 프로세스의 프로그램 명을 정확히 입력해야 한다. 유의할 점은 kill 명령은 프로세스 종료 신호(signal)을 전송하지만, 이것은 단지 해당 프로세스에게 종료를 요청하는 것이지 프로세스를 강제 종료하지 않는다는 점이다. 따라서, 확실하게 프로세스를 종료하고 싶은 경우에는 killp.sh -9 httpd와 같이 -9 인자를 추가해야 한다.

Source

```

#!/usr/bin/ksh
# -----
# 쉘 프로그램명: killp.sh
# 용도          : kill the named processes
#                shell script의 인자(arguments)로 지정한 이름을 가진 프로세스를
#                찾아 kill 명령을 수행한다.
# 사용법        :
#                killp.sh httpd
#                killp.sh -9 httpd
# -----

# 인자(argument)가 하나인 경우, 첫번째 인자는 프로세스 명
if [ $# == 1 ]
then
PROCESS="$1"

```

```
OPT=""
# 인자(argument)가 둘인 경우,
# 첫번째 인자는 kill 명령의 옵션, 두번째는 프로세스 명
elif [ $# == 2 ]
then
PROCESS="$2"
OPT="$1"

# 인자의 갯수가 틀린 경우는 '사용법'을 표시하고 종료한다.
else
echo Usage: killp [-sig] procname
exit 1
fi

# 종료되는 프로세스 갯수 계산
CNT=0

# ps 명령과 grep 명령을 조합한 후 파이프(pipe)를 이용하여
# 종료할 프로세스 목록을 추출한다.
/usr/bin/ps -eopid,comm | grep "$PROCESS" | while read pid cmd
do
f=`basename $cmd`
if [ $f = "$PROCESS" ]
then
echo $pid $cmd
CNT=1
kill $OPT $pid
fi
done

# 종료된 프로세스가 없다면, 오류 메시지 출력
if [ $CNT = 0 ]
then
echo $PROCESS not found
exit 0
fi
```

Arguments

1개인 경우, 종료 시키려는 프로세스 명

2개인 경우, 첫번째는 kill 명령에서 대한 옵션(option), 두번째는 종료 시키려는 프로세스명

Examples

killp.sh ns-httpd

killp.sh -9 uxwdog

3.2 fg (find and grep files)

Description

현재 디렉토리 하위의 모든 디렉토리에서 파일을 검색하고, 파일의 내용에 특정 문자열이 포함되어 있는지 검사하고 그 결과를 출력하는 스크립트이다.

Source

```
#!/bin/csh
```

```
# -----
# 쉘 프로그램명: fg.sh
# 용도          : find and grep files
#               shell script의 첫번째 인자로 지정한 이름을 가진 파일을
#               현재 디렉토리 아래에서 검색하고, 검색된 파일의 내용에 두번째 인자로
#               받은 문자열이 포함되어 있는지 검사한다.
# 사용법        :
#               fg.sh '*.txt' keyword
#               fg.sh README.txt year_month
# -----

find . -name "$1" -print -exec grep -i "$2" {} \;
echo
```

Arguments

인자의 개수는 2개이며, 생략해서는 안된다.

첫번째 인자는 검색하고자 하는 파일의 명칭이며, 파일 명칭이 불특정한 파일을 검색하고자 할 경우에는 wild character (*)를 사용할 수 있다. 단, wild character를 사용할 경우에는 첫번째 인자를 인용부호(quote)를 이용해 감싸주어야 한다. 두번째 인자는 검색된 파일 내의 내용을 검색하기 위한 키워드(keyword)이다.

Examples

아래 예는 현재 디렉토리 아래에서 확장자가 java인 파일들을 검색하고, 그 중에서 hello라는 문자열을 포함한 파일을 찾는 예제이다.

```
[zeus:/home1/work/public] fg.sh '*.java' hello
```

3.3 port_traffic (count network port traffic)

Description

TCP/IP 프로토콜 기반 서비스(http, telnet, FTP 서비스 등)를 운영하는 unix 서버에서 각 서버 프로세스 혹은 포트(port)에 매 순간 연결되어 있는 네트워크 연결 수 혹은 교통량(traffic load)을 측정하기 위한 스크립트이다. 이 스크립트를 통해 알 수 있는 수치는 대략적인 통신 부하 혹은 서버 부하량일 뿐 정확한 데이터라고 볼 수 없다. 네트워크 연결 정보 중에서 비정상적인 세션(session)이나, 통신 종료로 인해 대기 중인 세션들도 포함되어 연결 수가 계산되기 때문이다.

Source

```
#!/bin/csh
```

```
# -----
```



```
# 셸 프로그램명: port_traffic.sh
# 용도      : count network port traffic
#           각 서버 프로세스 혹은 포트(port)에 매 순간 연결되어 있는
#           네트워크 연결 수 혹은 교통량(traffic load)을 측정
# 사용법    :
#           port_traffic.sh bsw10.80
# -----

/usr/bin/netstat | grep "$1" | wc -l
```

Arguments

인자는 하나이며, 생략할 수 없다. 인자의 형식(format)은 ‘호스트명.포트번호’이다.

Examples

다음 예는 bsw10 서버에서 80 포트에 접속된 연결 수를 조회하는 예제이다.

```
[junet@bsw10:/etc]% port_traffic.sh bsw10.80
```

3.4 trace_mem (trace memory usage)

Description

시간의 변화와 프로세스 실행 상황에 따라 수시로 변화하는 메모리 사용 상태를 로그 파일에 기록하는 스크립트이다. 서버 상에서 동작 중인 모든 프로세스들의 실행 경로와 메모리 사용량을 kilo byte 단위로 정보를 출력한다.

Source

```
#!/bin/csh
# -----
# 셸 프로그램명: trace_mem.sh
# 용도      : 시간의 흐름에 따라 변화하는 메모리 사용 상태를 로그 파일에 기록.
#           현재 시스템에서 동작 중인 모든 프로세스들의 정보를 출력하며,
#           출력되는 정보의 종류는 다음과 같다.
#
#           vsz - Total size of the process in virtual memory
#           rss - The resident set size of the process
#           args - The command with all its arguments as a string.
#
# 사용법    : 프롬프트(prompt) 상에서 ./trace_mem.sh를 실행한다.
#           장시간 실행할 경우에 log-out 하게 되면 셸 프로그램이 종료되므로
#           로그 작성 중에는 log-out해서는 안된다..
# -----

# 최대 로그 작성 횟수. (1시간이면 60, 10시간이면 600)
set limit = 60
set cnt = 1

# 반복~ 또 반복..
while ( $cnt <= $limit )
```

```

# 로그 기록을 저장하는 텍스트 파일명 (현재, 일시분초를 이용해 파일명 생성)
set log_file_name = "ps_report`date +%d%H%M%S`.txt"

# 로그 파일을 생성한다.
touch $log_file_name

# 현재 시간을 기록한다.
echo - start ----- >> $log_file_name
date                                     >> $log_file_name

# 전체 메모리 사용량을 기록한다.
echo - vmstat ----- >> $log_file_name
vmstat                                >> $log_file_name

# 프로세스 상태를 기록한다. (자세한 정보는 man ps)
echo - process ----- >> $log_file_name
ps -aef -o vsz,rss,args                >> $log_file_name
echo - end ----- >> $log_file_name

# 기록 횟수를 증가 시키고~
@ cnt++

# 60초 후에 다시 기록한다..
sleep 60

end

```

Arguments

없음

Examples

[zeus:/home1/work/public/scripts] trace_mem.sh

위와 같이 실행하면 스크립트를 실행한 디렉토리에 다음 유형의 로그 파일들이 생성된다.

```

ps_report01071542.txt
ps_report01071642.txt
ps_report01071742.txt
ps_report01071842.txt
.....

```

3.5 mon_cpu (monitor CPU load)

Description

CPU의 부하를 측정하는 스크립트. 서버 프로세스 중에서 가장 부하가 높은 10개의 프로세스 목록을 보여준다.

Source

```
#!/bin/ksh
```

```

while
true
do
    clear
    echo "==== current system load ==="
        uptime
    echo " "
    echo " "
    echo "=== top 10 processes using CPU === "
    echo "          |--- CPU usage"
    echo "          v "
        ps -efa | /usr/bin/sort -n -r +3 | head -10
    sleep 5
done

```

Arguments

없음

Examples

없음.

3.6 zip_n_ext (zip or extract files)

Description

현재 디렉토리 이하의 모든 파일을 tar 및 gzip 명령을 이용해 결합하고, 압축한 후 인자로 지정된 파일명으로 저장하거나 혹은 지정된 압축 파일을 해제하여 현재 디렉토리 아래에서 위치시킨다. 압축 시에는 현재 디렉토리 하위의 모든 파일을 압축한 후 인자로 지정된 파일명으로 저장하고, 해제 시에는 인자로 지정된 파일을 해제한다. 즉, 선택적으로 압축하거나 해제하는 기능은 제공되지 않는다.

Source

```

#!/bin/ksh
# -----
# 쉘 프로그램명: zip_n_ext.sh
# 용도          : 현재 디렉토리 하위의 모든 파일을 압축하거나
#                  압축된 파일을 해제한다~
# 사용법        : zip_n_ext c backup
#                  zip_n_ext d archive
# -----

# 명령 인자 갯수가 틀린 경우...
if [ $# != 2 ]
then
echo Usage: zip_n_ext.sh [c/d] zipfile
exit 1
fi

```

```

# 압축한다~~
if [ $1 == "c" ]
then
ZIP_FILE_NAME="$2"
ACT="decompress"

# 해제한다~~
elif [ $1 == "d" ]
then
    ZIP_FILE_NAME="$2"
    ACT="compress"

# 명령 형식이 틀린 경우..
else
echo Usage: zip_n_ext.sh [c/d] zipfile
exit 1
fi

if [ $ACT == "compress" ]
then
    tar cf - * | gzip > $ZIP_FILE_NAME.tgz
else
    gzip -cd $ZIP_FILE_NAME.tgz | tar xf -
fi

```

Arguments

실행 시 인자는 항상 2개이며, 생략할 수 없다.

첫번째 인자는 압축할 것인지 혹은 해제할 것인지를 나타내는 작업 선택으로 압축인 경우에는 'c', 해제인 경우에는 'd'를 입력한다. 두번째 인자는 압축할 때는 통합되는 파일의 명칭이며, 해제될 경우에는 압축 파일의 명칭으로 확장자는 제외하고 입력하여야 한다.

Examples

다음은 현재 디렉토리 아래의 모든 파일을 압축한 후, backup.tgz 파일을 생성한다.

```
[zeus:/home1/work/public/scripts] zip_n_ext c backup
```

다음은 archive.tgz 파일을 해제하여 현재 디렉토리에 파일들을 위치 시킨다.

```
[zeus:/home1/work/public/scripts] zip_n_ext d archive
```

3.7 restart_all_websvr (restart all web servers)

Description

개발용 혹은 운영 시스템에서 웹 서버 인스턴스(instance)를 복수로 운영할 때, 서버의 설정 변경, 라이브러리의 컴파일, 데이터베이스의 재기동 등의 작업으로 인해 모든 서버 인스턴스를 일시에 중단시킨 후, 기동시키고자 할 때 사용된다.

Source

```
#!/bin/csh
```

```
set HOME = /home1/work/iplanet/https-

foreach server_id ( admserv zeus bbs beta comm infomian kms mail pims
portal request sign support)
  cd $HOME$server_id

  switch ( $1 )
    case 'start':
      ./start
      breaksw
    case 'stop':
      ./stop
      breaksw
    case 'restart':
      ./stop
      ./start
      breaksw
    default:
      echo 'input error'
      breaksw
  endsw
end
```

Arguments

없음

Examples

없음

3.8 oracle_start8 (startup oracle 8)

Description

oracle 8 버전 서버를 기동하는 스크립트.

Source

```
#!/bin/sh -f

# description - Oracle startup script for ORACLE7 or ORACLE8
# caution - Do not use this script for ORACLE9 or upper version

svrmgrl << EOF
connect internal
startup
EOF
lsnrctl start
```

Arguments

없음

Examples

없음

3.9 oracle_stop8(shutdown oracle 8)

Description

oracle 7 혹은 8 버전 서버를 종료하는 명령

Source

```
#!/bin/sh -f
lsnrctl stop
svrmgrl << EOF
connect internal
shutdown abort
EOF
```

Arguments

없음

Examples

없음

3.10 oracle_start9 (startup oracle 9)

Description

oracle 9 버전 서버 기동(startup)

Source

```
#!/bin/ksh -f

# description - Oracle startup script for ORACLE9
# caution - Do not use this script for ORACLE8 or lower version

sqlplus "/as sysdba" << EOF
startup
EOF
lsnrctl start
```

Arguments

없음

Examples

없음

3.11 oracle_stop9 (shutdown oracle 9)

Description

oracle 9 버전 서버를 종료하는 스크립트

Source

```
#!/bin/ksh -f

lsnrctl stop
sqlplus "/as sysdba" << EOF
shutdown immediate
EOF
```

Arguments

없음

Examples

없음

3.12 restart_tomcat

Description

톰캣 서버를 재시작하는 스크립트

Source

```
#!/bin/csh

pushd .
cd $CATALINA_HOME/bin
shutdown.sh
sleep 1
startup.sh
popd
```

Arguments

없음

Examples

없음

3.13 mon_log (monitor log)

Description

인자(parameter)로 지정한 로그 파일의 최근 50개의 행(row)을 모니터하는 스크립트

Source

```
#!/bin/csh

set n = 1

while ($n <= 10000 )
```

```
do
    tail -n 50 "$1"
    sleep 5
done
```

Arguments

인자는 하나이며, 생략할 수 없다. 모니터링 하고자 하는 로그 파일명이다.

Examples

없음