

# AIX ?????

??????

```
# Login Failed
who /etc/security/failedlogin | tail -50

# Check the number of previous unsuccessful logins for the account to confirm it is blocked
lsuser -a account_locked unsuccessful_login_count {ALL|user_name}

# Reset unsuccessful login counter
chsec -f /etc/security/lastlog -a unsuccessful_login_count=0 -s {user_name}

# Unlock the locked account
chuser account_locked=false {user_name}

# Lock account
chuser account_locked=true {user_name}
```

## ?? errpt

Sample #1

```
#!/usr/bin/env bash
#
# $0 = errptcheck_v3.sh
#
# Created: 05/16/2005 A-lang Hsu.
# Updated:
#   - v4, 11/19/2015 A-Lang
#   - v5, 11/17/2020 A-Lang
#
#
# This script will check the error log
# for new entries. Upon finding them, it will send an email to
# administrators containing a message indicating the change
```

```
# in errlog status, as well as the offending lines.

#
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/es/sbin/cluster/utilities

tmpfile="errptcheck.$$"
trap "rm -f $tmpfile" EXIT

today="$(date +%Y-%m-%d)"
nowtime="$(date +%T)"

# Excluded identifier id
# 573790AA - The default log file has been changed.
# A3B02BE6 - sddsrv CAN'T WRITE ITS LOG FILE
#
#EXCLUDE_ID=" \
#573790AA \
#A3B02BE6 \
#"
EXCLUDE_ID=""

#
if $(which get_local_nodename >/dev/null 2>&1); then
    my_hostname=`hostname`#`get_local_nodename`
else
    my_hostname=`hostname`
fi

mail_subject="Warning:Hardware/Software error notification for host $my_hostname"
mail_to="alang@mycom.com"

#ec=`errpt -dH,S,U,O | grep -v "IDENTIFIER TIMESTAMP" | wc -l`
ERRGREP=""
n=0
for i in $EXCLUDE_ID;do
    n=$((n+1))
    if [ "$n" -eq 1 ];then
        ERRGREP="$i"
    else
        ERRGREP="$ERRGREP|$i"
    fi
```

```

done
if [ -z "$ERRGREP" ]; then
    ec=`errpt -dH,S,U,O | grep -v "IDENTIFIER TIMESTAMP" | wc -l`
else
    ec=`errpt -dH,S,U,O | grep -v "IDENTIFIER TIMESTAMP" | grep -vE "$ERRGREP" | wc -l`
fi

if [ "$ec" -ne "0" ]; then
    ec=`echo $ec | bc`
    cat <<EOF > $tmpfile
#####
    This message was generated automatically by host ${my_hostname}.
    Please don't reply to this message.
#####

    The checked time is ${today} ${nowtime}
    ${ec} new errors have been found on $my_hostname.
    EOF

    errlogl=`errpt -dH,S,U,O -a`
    cat <<EOF >> $tmpfile
Errlog details below:
${errlogl}
EOF
    mail -s "$mail_subject" "$mail_to" < $tmpfile;

    cat $tmpfile
fi

```

## Sample #2

```

#!/bin/ksh
#
# $0 = errmon.sh
#
# Written 11/3/1998 Bill Verzal.
#
# This script will run every [interval] and check the error log
# for new entries. Upon finding them, it will send an email to
# administrators containing a message indicating the change

```

```

# in errlog status, as well as the offending lines.
#
if [ "$1" = "-v" ] ; then
    set -x
fi
lc="NULL"
tc="$lc"
# lc="last count"
# tc="this count"
#interval=900
interval=300
# Divide interval by 60 to get number of minutes.
me="$0 - Hardware error monitoring"
myname=`hostname`
args="$*"
#mailto="root"
mailto="alert"
true=0
false=1
boj=`date`

echo "$me started.\nThis message goes to $mailto." | mail -s "Errlog monitoring for $myname" $mailto
logger "$0 started"

while [ "$true" != "$false" ] ; do
    tc=`errpt -dH,S,U,O | wc -l`
    if [ "$lc" = "NULL" ] ; then
        lc="$tc"
    fi
    if [ "$lc" -ne "$tc" ] ; then
        foo=`echo "$tc-$lc"|bc`
        msg="$foo new errors have been found on $myname"
        page_msg="$foo new errors have been found on $myname"
        errlogl=`errpt -dH,S,U,O -a`
        if [ "$tc" -eq "0" ] ; then
            msg="$msg\n Errlog was cleared"
        else
            logger $msg
            msg=" $msg \n Errlog details below:\n $errlogl \n"
            echo "$msg" | mail -s "Errlog status change on host $myname" $mailto
        fi
    fi
done

```

```
fi
fi
lc="$tc"
sleep $interval
done
```

## ?? HMC root

- [?? HMC8 ? HMC9 ? root ??](#)

## Restricted Shell

???????????????? Shell ????????

???

- [How to Use a Restricted Shell](#)

### Default Shell?


```
# Change the default shell for the user to the restricted shell such as rksh or Rsh.
chuser shell=/usr/bin/rksh <user-name>
# OR
chsh <user-name> /usr/bin/rksh
```

### .profile?

```
# Add the commands that are allowed to run by the user into the directory.
mkdir /usr/bin/restricted
cd /usr/bin/restricted
ln -s /usr/bin/date date

# Create a .profile in the user's home directory and set the PATH environment variable to
# a directory containing all of the commands you want the user to be able to run
export PATH=/usr/bin/restricted
```

## Core dump

```
#  core file
dbx -C ./core
```

```
(dbx) corefile
```

```
(dbx) dump
```

```
(dbx) quit
```

## System dump

errpt:

```
67145A39 0413095315  U  S  SYSDUMP  SYSTEM DUMP
```

Copy the dump from the dump device to a file using the `savecore` command:

```
savecore .
```

Yes, the period is necessary. It indicates you want the dump copied to your current directory

`savecore` will copy the dump to your current directory, and name it:

```
vmcore.0.BZ
```

Uncompress the dump using the `dmpuncompress` command:

```
dmpuncompress vmcore.0.BZ
```

Lastly, format the dump:

```
/usr/lib/ras/dmptns/dmpfmt -c vmcore.0
```

## Reading a Dump

```
kdb vmcore.0 vmunix.0
```

????

Memory - svmon

```
# For a summary of the top 15 processes using memory on the system
svmon -Pt15 | perl -e 'while(<>){print if($.==2||$&&&!$s++);$.=0 if(/^-$$/)}'
```

```
-----
  Pid Command      Inuse   Pin   Pgspace Virtual 64-bit Mthrd 16MB
18547096 db2sysc    3956861 12944 282407 4007901   Y   Y   N
19333470 db2sysc     690873 12944 26772 688572   Y   Y   N
19726694 db2sysc     271696 12944 6198 287133   Y   Y   N
13500914 db2sysc     263458 12943 18957 285159   Y   Y   N
 1966448 shlap64    109377 12900 3432 122071   Y   N   N
13631924 db2vend     105589 12900 597 115784   Y   N   N
19005734 db2sysc    105082 12902 409 114965   Y   Y   N
20709798 db2sysc    105071 12900 409 114953   Y   N   N
20119938 db2sysc    105071 12900 409 114953   Y   N   N
20185458 db2sysc    105071 12900 408 114953   Y   N   N
15597848 db2vend     104222 12900 1771 115608   Y   N   N
21430722 db2sysc    103728 12900 1576 114777   Y   N   N
21037528 db2sysc    103724 12902 1576 114773   Y   Y   N
14025064 db2sysc    103696 12900 1608 114777   Y   N   N
18350424 db2sysc    103696 12900 1608 114777   Y   N   N
```

## Perl ??

??????????

```
perl -e "use LWP::UserAgent;"
perl -e "use DBI;"
```

## HTTP GET request

```
use LWP::UserAgent;

my $ua = LWP::UserAgent->new;

my $server_endpoint = "http://192.168.1.1:8000/service";

# set custom HTTP request header fields
my $req = HTTP::Request->new(GET => $server_endpoint);
$req->header('content-type' => 'application/json');
$req->header('x-auth-token' => 'kfksj48sdfj4jd9d');
```

```

my $resp = $ua->request($req);
if ($resp->is_success) {
    my $message = $resp->decoded_content;
    print "Received reply: $messagen";
}
else {
    print "HTTP GET error code: ", $resp->code, "n";
    print "HTTP GET error message: ", $resp->message, "n";
}

```

## HTTP POST request

```

use LWP::UserAgent;

my $ua = LWP::UserAgent->new;

my $server_endpoint = "http://192.168.1.1:8000/service";

# set custom HTTP request header fields
my $req = HTTP::Request->new(POST => $server_endpoint);
$req->header('content-type' => 'application/json');
$req->header('x-auth-token' => 'kfksj48sdfj4jd9d');

# add POST data to HTTP request body
my $post_data = '{ "name": "Dan", "address": "NY" }';
$req->content($post_data);

my $resp = $ua->request($req);
if ($resp->is_success) {
    my $message = $resp->decoded_content;
    print "Received reply: $messagen";
}
else {
    print "HTTP POST error code: ", $resp->code, "n";
    print "HTTP POST error message: ", $resp->message, "n";
}

```

## AIX Toolbox



- [AIX Toolbox for Open Source Software : Downloads alpha \(ibm.com\)](#)
- [Configuring YUM and creating local repositories on IBM AIX - IBM Developer](#)

---

Revision #15

Created 23 December 2020 06:24:47 by Admin

Updated 19 March 2024 15:39:53 by Admin