If I Send a Special Message to Linux Will it Answer?

Session 12090

Richard F. Lewis

IBM Corp.

rflewis@us.ibm.com

Trademarks

Agenda

- smsgiucv_app
- z/VM SMSG
- Udev
- Lots of examples
- CDSL
- Summary

smsgiucv_app Device Driver

- Kernel device driver that receives CP special messages
- Delivers messages to user space as udev events
- Messages must begin with identifier APP
- Environment variables deliver content to user space programs
- Linux prerequisites
  - Redhat Enterprise Linux 6.1 or later
  - SUSE Linux Enterprise Server 11 SP2 or later

smsgiucv_app Device Driver

LNX001

application

udev rule → udevd

SECOPER

```
CP SMSG LNX001 APP hi
```

```
UEVENT
ACTION=change
SMSG_SENDER=SECOPER
SMSG_ID=APP
SMSG_TEXT=hi
```

Linux kernel

```
smsgiucv_app
device driver
```

CP
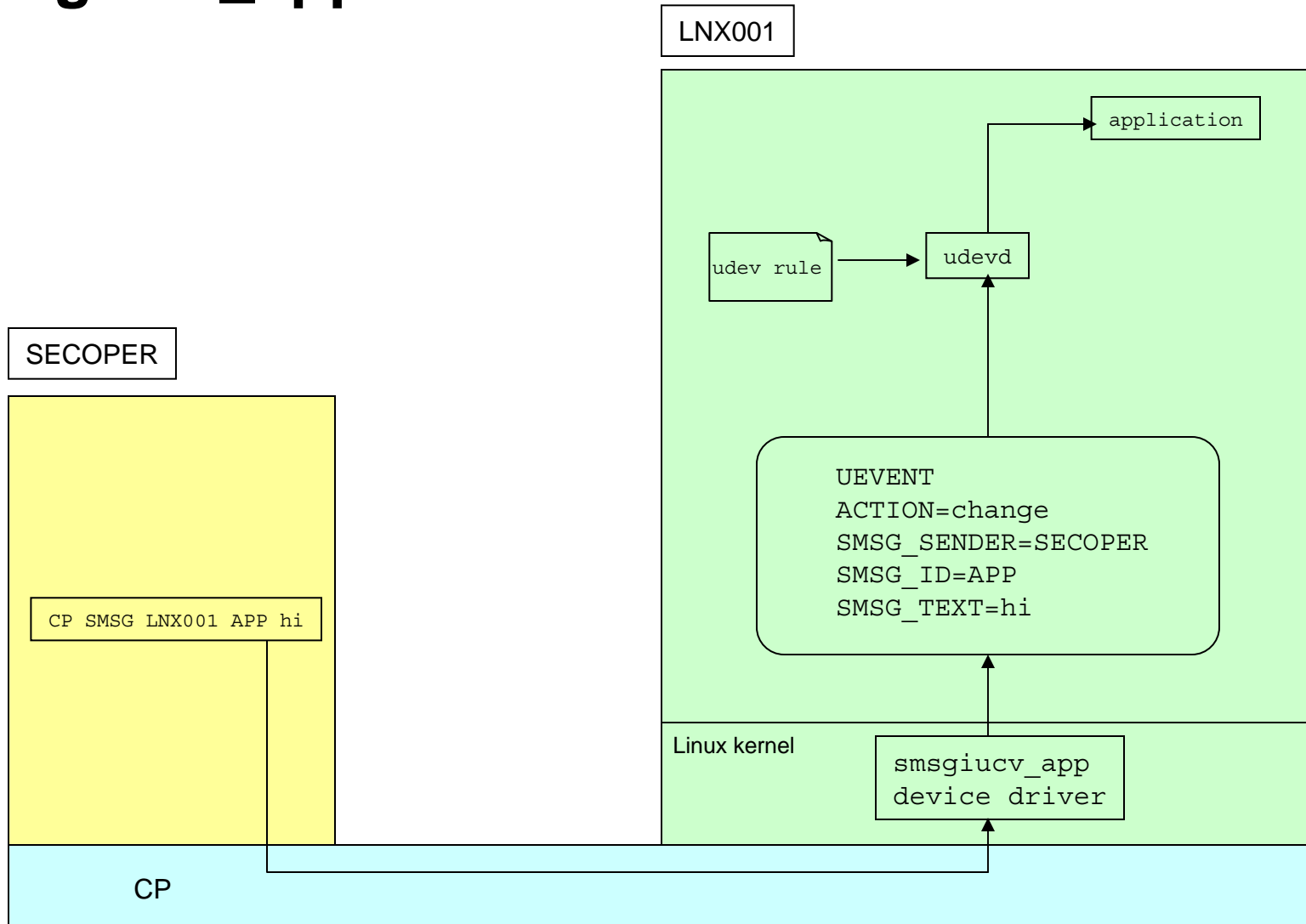
What is SMSG?

- z/VM SMSG Facility
  - Class G CP Command
  - Pass message to program executing in another virtual machine
  - Used with VMCF or IUCV protocols
  - Target virtual machine must:
    - Enable for external interrupts
    - Set bit 30 of CR 0 to a value of 1
    - Issue IUCV DECLARE BUFFER
    - Issue IUCV CONNECT to CP MSG System Service
    - SET SMSG IUCV (turn on special message flag)

What is Udev?

- Udev developed to dynamically create device nodes in /dev
  - Replace static device definitions
    - All devices that might appear
  - Only create definitions for devices that exist
  - Create block special and persistent names
  - Powerful and flexible facility
- User space tool
- Daemon facility and <u>rules</u> facility

UDEV Rules

- Constructed from comma separated key=value pairs
- Match keys identify device the rule acts upon
  - All match keys in a rule must match to apply rule
- Assignment keys set and other actions occur when rule applied
- All matching rules in rules database applied
- Rules applied in lexical order

UDEV Rules

- A few common match keys
  - *Kernel*: match against kernel name for device
  - *Subsystem*: match against subsystem of device
  - *Driver*: match against name of driver backing device
  - *Action*: match against name of event action*
  - *Devpath*: match devpath of event device

  * Used commonly with smsgiucv_app driver

UDEV Rules

- A few common assignment keys
  - *Name*: node to be created
  - *Symlink*: name of symlink targeting node
  - *ATTR{key}:* set value of sysfs attribute of event device
  - *Run*: list of programs to be executed for device *
  - *Label*: target for a goto jump *
  - *Goto*: jump to label with matching name *
  - *Program*: run an external program to produce a device name *

  * Used with smsgiucv_app driver

UDEV Rules

- Environment variables
  - Available to programs executed by rules
  - Can be set by rules
  - *SMSG_ID*: message prefix set to APP *
  - *SMSG_SENDER*: virtual machine that sent message *
  - *SMSG_TEXT*: text of the CP SMSG *
  - Result: output from external program run to produce a device name

* Unique to smsgiucv_app driver

UDEV Rules

- ## String matching

  - * - match any character, zero or more times

  - ? – match any character exactly once

  - [] – match any single character specified in the brackets, ranges are supported

UDEV Rules

- Key operations
  - **==** compare for equality
  - **!=** compare for inequality
  - **=** assign a value to a key
  - **+=** add a value to a key that holds a list of entries
  - **:=** add a final value to a key holding a list of entries

UDEV Rules

- **String substitutions**
  - *$kernel or %k* – kernel name for this device
  - *$devpath or %p* – devpath for this device
  - *$env{key} or %E{key}* – device property value
  - *$major or %M* – device major number
  - *$minor or %m* – device minor number
  - *$result or %c* – string returned by program executed through PROGRAM statement

UDEV Actions

- *ADD* – sent when device added or smsgiucv_app device driver is loaded

- *REMOVE* – sent when device removed or smsgiucv_app device driver is unloaded

- *CHANGE* – sent by smsgiucv_app device driver when an SMSG is received (not as common an action with devices)

Utilities

- udevadm – udev management tool
  - Obtain information from /sys tree
  - Test rules
  - Trigger processing (coldplug)
  - Monitor event processing and timing
  - Control internal state of udev daemon (reload rules etc)

UDEVADM

```
udevadm info --attribute-walk --path=/devices/css0/0.0.000b/0.0.0200
```

```
Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.
```

```
  looking at device '/devices/css0/0.0.000b/0.0.0200':
    KERNEL=="0.0.0200"
    SUBSYSTEM=="ccw"
    DRIVER=="dasd-eckd"
    ATTR{devtype}=="3390/0a"
    ATTR{cutype}=="3990/e9"
    ATTR{modalias}=="ccw:t3990mE9dt3390dm0A"
    ATTR{online}=="1"
    ATTR{cmb_enable}=="0"
    ATTR{availability}=="good"
    ATTR{readonly}=="0"
    ATTR{discipline}=="ECKD"
    ATTR{status}=="online"
    ATTR{alias}=="0"
    ATTR{vendor}=="IBM"
    ATTR{uid}=="IBM.750000000FBKX1.18f1.60.0000000100000d0a0000000000000000"
    ATTR{use_diag}=="0"
    ATTR{raw_track_access}=="0"
    ATTR{eer_enabled}=="0"
    ATTR{erplog}=="0"
    ATTR{failfast}=="0"
    ATTR{expires}=="30"
    ATTR{failfast_expires}=="30"
    ATTR{retries}=="256"
    ATTR{failfast_retries}=="256"
    ATTR{reservation_policy}=="ignore"
    ATTR{last_known_reservation_state}=="none"
```

```
  looking at parent device '/devices/css0/0.0.000b':
    KERNELS=="0.0.000b"
    SUBSYSTEMS=="css"
    DRIVERS=="io_subchannel"
    ATTRS{type}=="0"
    ATTRS{modalias}=="css:t0"
    ATTRS{chpids}=="88 89 8a 8b 8c 8d 8e 8f "
    ATTRS{pimpampom}=="ff ff ff"

  looking at parent device '/devices/css0':
    KERNELS=="css0"
    SUBSYSTEMS==""
    DRIVERS==""
```

```
udevadm info --attribute-walk --path=/devices/iucv/smsgiucv_app

Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

  looking at device '/devices/iucv/smsgiucv_app':
    KERNEL=="smsgiucv_app"
    SUBSYSTEM=="iucv"
    DRIVER=="SMSGIUCV"

  looking at parent device '/devices/iucv':
    KERNELS=="iucv"
    SUBSYSTEMS==""
    DRIVERS==""
```

Getting Started

- First line and last line should look like following
  - Allows udevd to skip rules if they are not related to smsgiucv_app
  - LABEL matchkey is at the end of your file

```
DEVPATH!="*/smsgiucv_app", GOTO="smsgiucv_app_end
  <lots of rules go here>
LABEL="smsgiucv_app_end"
```

Device Driver Loading

- When the smsgiucv_app device driver is loaded it sends an ADD udev event
- This allows for rules to be defined that load dependent modules or prepare the environment in some other way.

```
SUBSYSTEM=="module", ACTION=="add", RUN+="/sbin/modprobe --quiet vmcp"
```

```
SUBSYSTEM=="module", ACTION=="add", RUN+="/sbin/modprobe --quiet vmcp"
SUBSYSTEM=="module", ACTION=="add", RUN+="/sbin/vmcp msg SECOPER APP AVAIL"
SUBSYSTEM=="module", ACTION=="remove", RUN+="/sbin/vmcp msg SECOPER APP DOWN"
```

Simple Processing

- Message response with uname

```
ACTION=="change", ENV{SMSG_TEXT}=="HI", \
    PROGRAM=="/bin/uname -n -r", \
    RUN+="/sbin/vmcp msg $env{SMSG_SENDER}'$result'"
```

```
sm sls11sp2 APP hi
Ready; T=0.01/0.01 16:25:19
 16:25:19  * MSG FROM SLS11SP2: SLS11SP2 3.0.13-0.23-DEFAULT
```

- PROGRAM statement runs uname command
- VMCP sends $result back to sender

PROGRAM

- Use PROGRAM key to execute program to name device
    - Output of program is in result environment variable
    - Program being executed needs to be very short running
    - Only one key per rule
    - No default path available
    - In context of smsgiucv_app reacts to smsg

PROGRAM Example

- Pearl script to print it's environment vars

```
ACTION=="change", ENV{SMSG_TEXT}=="PE", \
   PROGRAM=="/sbin/printenv.pl"
```

```
sm sls11sp2 APP PE
Ready; T=0.01/0.01 13:36:01
  13:36:01  * MSG FROM SLS11SP2: ACTION=CHANGE
  13:36:01  * MSG FROM SLS11SP2: DEVPATH=/DEVICES/IUCV/SMSGIUCV_APP
  13:36:01  * MSG FROM SLS11SP2: DRIVER=SMSGIUCV
  13:36:01  * MSG FROM SLS11SP2: SEQNUM=491
  13:36:01  * MSG FROM SLS11SP2: SMSG_ID=APP
  13:36:01  * MSG FROM SLS11SP2: SMSG_SENDER=MAINT
  13:36:01  * MSG FROM SLS11SP2: SMSG_TEXT=PE
  13:36:01  * MSG FROM SLS11SP2: SUBSYSTEM=IUCV
  13:36:01  * MSG FROM SLS11SP2: UDEV_LOG=3
```

```
#!/usr/bin/perl
foreach $key (sort keys(%ENV))
{
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "$key=$ENV{$key}\n";
}
```

PROGRAM Example

- Setting PATH

```
ACTION=="change", ENV{SMSG_TEXT}=="SHOWPATH", \
   PROGRAM=="/sbin/mysetpath", \
   RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
sm sls11sp2 APP showpath
Ready; T=0.01/0.01 13:47:04
 13:47:04  * MSG FROM SLS11SP2: PATH STARTS AS /USR/LOCAL/BIN:/BIN:/USR/BIN:/USR/X11R6/BIN:. PATH IS NOW
/SBIN:/USR/LOCAL/BIN:/BIN:/USR/BIN:/USR/X11R6/BIN:.
```

```
cat /sbin/mysetpath
#!/bin/sh
echo "Path starts as" $PATH
export PATH=/sbin:$PATH
echo "Path is now" $PATH
```

RUN

- List of programs to be executed for a specific device
- Should be short running programs
- Same environment variables available as with PROGRAM
- Limited path
- Does not assign anything to result environment variable

RUN Example

- Execute vmcp to return content of result variable to sender

```
ACTION=="change", ENV{SMSG_TEXT}=="HI", \
    PROGRAM=="/bin/uname -n -r", \
    RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
sm sls11sp2 APP HI
Ready; T=0.01/0.01 14:08:59
 14:08:59   * MSG FROM SLS11SP2: SLS11SP2 3.0.13-0.23-DEFAULT
```

RUN Example

- Multiple programs in list

```
ACTION=="change", ENV{SMSG_TEXT}=="HI2", \
    PROGRAM=="/bin/uname -n -r", \
    RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'", \
    RUN+="/sbin/vmcp msg $env{SMSG_SENDER} APP1", \
    RUN+="/sbin/vmcp msg $env{SMSG_SENDER} APP2", \
    RUN+="/sbin/myg $env{SMSG_SENDER} "
```

```
sm sls11sp2 APP HI2
Ready; T=0.01/0.01 14:44:36
 14:44:36  * MSG FROM SLS11SP2: SLS11SP2 3.0.13-0.23-DEFAULT
 14:44:36  * MSG FROM SLS11SP2: APP1
 14:44:36  * MSG FROM SLS11SP2: APP2
 14:44:36  * MSG FROM SLS11SP2: SLS11SP2 3.0.13-0.23-DEFAULT
```

```
#!/bin/sh
uid=$1
shift
rest=$@
/sbin/vmcp msg $uid $rest
```

RUN Example

```
sm sls11sp2 APP mem
Ready; T=0.01/0.01 14:50:23
 14:50:23  * MSG FROM SLS11SP2:                   TOTAL       USED       FREE     SHARED    BUFFERS     CACHED
 14:50:23  * MSG FROM SLS11SP2: MEM:            1020416     425856     594560          0       2108     369556
 14:50:23  * MSG FROM SLS11SP2: -/+ BUFFERS/CACHE:          54192     966224
 14:50:23  * MSG FROM SLS11SP2: SWAP:                 0          0          0
 14:50:23  * MSG FROM SLS11SP2: ----------------------
 14:50:23  * MSG FROM SLS11SP2: MEMTOTAL:           1020416 KB
 14:50:23  * MSG FROM SLS11SP2: MEMFREE:             594364 KB
 14:50:23  * MSG FROM SLS11SP2: BUFFERS:               2108 KB
 14:50:23  * MSG FROM SLS11SP2: CACHED:              336508 KB
 14:50:23  * MSG FROM SLS11SP2: SWAPCACHED:               0 KB
 14:50:23  * MSG FROM SLS11SP2: ACTIVE:              198364 KB
 14:50:23  * MSG FROM SLS11SP2: INACTIVE:            161348 KB
 14:50:23  * MSG FROM SLS11SP2: ACTIVE(ANON):         21252 KB
 14:50:23  * MSG FROM SLS11SP2: INACTIVE(ANON):          60 KB
 14:50:23  * MSG FROM SLS11SP2: ACTIVE(FILE):        177180 KB
 14:50:23  * MSG FROM SLS11SP2: INACTIVE(FILE):      161288 KB
 14:50:23  * MSG FROM SLS11SP2: UNEVICTABLE:              0 KB
 14:50:23  * MSG FROM SLS11SP2: MLOCKED:                  0 KB
 14:50:23  * MSG FROM SLS11SP2: SWAPTOTAL:                0 KB
 14:50:23  * MSG FROM SLS11SP2: SWAPFREE:                 0 KB
 14:50:23  * MSG FROM SLS11SP2: DIRTY:                    4 KB
 14:50:23  * MSG FROM SLS11SP2: WRITEBACK:                0 KB
 14:50:23  * MSG FROM SLS11SP2: ANONPAGES:            21120 KB
 14:50:23  * MSG FROM SLS11SP2: MAPPED:               10640 KB
 14:50:23  * MSG FROM SLS11SP2: SHMEM:                  148 KB
 14:50:23  * MSG FROM SLS11SP2: SLAB:                 42044 KB
 14:50:23  * MSG FROM SLS11SP2: SRECLAIMABLE:         33048 KB
 14:50:23  * MSG FROM SLS11SP2: SUNRECLAIM:            8996 KB
 14:50:23  * MSG FROM SLS11SP2: KERNELSTACK:           2000 KB
 14:50:23  * MSG FROM SLS11SP2: PAGETABLES:             684 KB
 14:50:23  * MSG FROM SLS11SP2: NFS_UNSTABLE:             0 KB
 14:50:23  * MSG FROM SLS11SP2: BOUNCE:                   0 KB
 14:50:23  * MSG FROM SLS11SP2: WRITEBACKTMP:             0 KB
 14:50:23  * MSG FROM SLS11SP2: COMMITLIMIT:         510208 KB
 14:50:23  * MSG FROM SLS11SP2: COMMITTED_AS:         81332 KB
 14:50:23  * MSG FROM SLS11SP2: VMALLOCTOTAL:     134217728 KB
 14:50:23  * MSG FROM SLS11SP2: VMALLOCUSED:          13936 KB
 14:50:23  * MSG FROM SLS11SP2: VMALLOCCHUNK:     134196180 KB
 14:50:23  * MSG FROM SLS11SP2: HUGEPAGES_TOTAL:          0
 14:50:23  * MSG FROM SLS11SP2: HUGEPAGES_FREE:           0
 14:50:23  * MSG FROM SLS11SP2: HUGEPAGES_RSVD:           0
 14:50:23  * MSG FROM SLS11SP2: HUGEPAGES_SURP:           0
 14:50:23  * MSG FROM SLS11SP2: HUGEPAGESIZE:          1024 KB
```
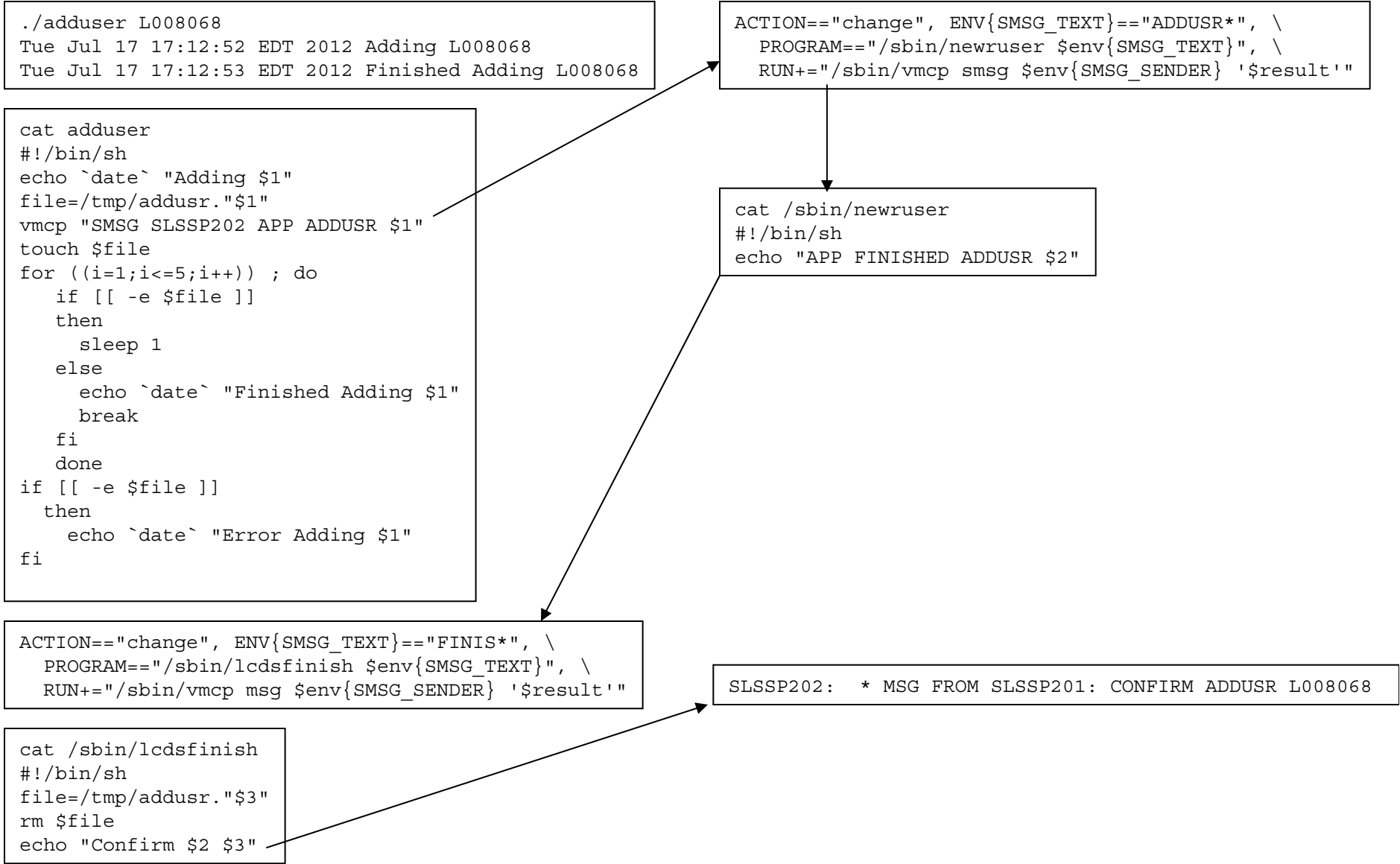
```
ACTION=="change", ENV{SMSG_TEXT}=="MEM*", \
  RUN+="/sbin/printfree.pl", \
  RUN+="/sbin/printproc.pl"
```

```
cat /sbin/printfree.pl
#!/usr/bin/perl
open STATUS, "/usr/bin/free 2>&1 |";
while (<STATUS>)
{
my($line) = $_;
chomp($line);
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "$line\n";
}
close STATUS;
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "----------------------\n";
```

```
cat /sbin/printproc.pl
#!/usr/bin/perl
open ( FILE, "/proc/meminfo" );
while (<FILE>)
{
my($line) = $_;
chomp($line);
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "$line\n";
}
close FILE;
```

Pseudo Conversation

```
./adduser L008068
Tue Jul 17 17:12:52 EDT 2012 Adding L008068
Tue Jul 17 17:12:53 EDT 2012 Finished Adding L008068
```

```
ACTION=="change", ENV{SMSG_TEXT}=="ADDUSR*", \
   PROGRAM=="/sbin/newruser $env{SMSG_TEXT}", \
   RUN+="/sbin/vmcp smsg $env{SMSG_SENDER} '$result'"
```

```
cat adduser
#!/bin/sh
echo `date` "Adding $1"
file=/tmp/addusr."$1"
vmcp "SMSG SLSSP202 APP ADDUSR $1"
touch $file
for ((i=1;i<=5;i++)) ; do
   if [[ -e $file ]]
   then
     sleep 1
   else
     echo `date` "Finished Adding $1"
     break
   fi
   done
if [[ -e $file ]]
   then
     echo `date` "Error Adding $1"
fi
```

```
cat /sbin/newruser
#!/bin/sh
echo "APP FINISHED ADDUSR $2"
```

```
ACTION=="change", ENV{SMSG_TEXT}=="FINIS*", \
   PROGRAM=="/sbin/lcdsfinish $env{SMSG_TEXT}", \
   RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
SLSSP202:  * MSG FROM SLSSP201: CONFIRM ADDUSR L008068
```

```
cat /sbin/lcdsfinish
#!/bin/sh
file=/tmp/addusr."$3"
rm $file
echo "Confirm $2 $3"
```

CMM

```
ACTION=="change", ENV{SMSG_TEXT}=="CMM*", \
    RUN+="/sbin/docmm.pl $env{SMSG_TEXT}"
```

```
sm slssp201 app cmm 512
Ready; T=0.01/0.01 10:13:29
 10:13:29  * MSG FROM SLSSP201:             TOTAL        USED        FREE      SHARED     BUFFERS      CACHED
 10:13:29  * MSG FROM SLSSP201: MEM:      1020416      148176      872240           0       17260       83408
 10:13:29  * MSG FROM SLSSP201: -/+ BUFFERS/CACHE:       47508      972908
 10:13:29  * MSG FROM SLSSP201: SWAP:           0           0           0
 10:13:30  * MSG FROM SLSSP201:             TOTAL        USED        FREE      SHARED     BUFFERS      CACHED
 10:13:30  * MSG FROM SLSSP201: MEM:      1020416      673272      347144           0       17260       83408
 10:13:30  * MSG FROM SLSSP201: -/+ BUFFERS/CACHE:      572604      447812

 10:13:30  * MSG FROM SLSSP201: SWAP:           0           0           0
```

```
cat /sbin/docmm.pl
#!/usr/bin/perl
open STATUS, "/usr/bin/free 2>&1 |";
while (<STATUS>)
{
my($line) = $_;
chomp($line);
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "$line\n";
}
close STATUS;
$p=(($ARGV[1]*1048576)/4096);
`echo $p > /proc/sys/vm/cmm_pages`;
`sleep 1`;
open STATUS, "/usr/bin/free 2>&1 |";
while (<STATUS>)
{
my($line) = $_;
chomp($line);
system "/sbin/vmcp", "MSG $ENV{SMSG_SENDER}", "$line\n";
}
close STATUS;
```

Suspend

```
ACTION=="change", ENV{SMSG_TEXT}=="SUSP*", \
  RUN+="/sbin/dosusp"
```

```
 <----- CPU Load -----> <------ Virtual IO/s ------>                      <-User Time-> <--Spool-->    MDC
               <-Seconds->   T/V                                          <--Minutes--> Total  Rate Insert
 Userid    %CPU  TCPU  VCPU Ratio Total DASD Avoid Diag98   UR Pg/s User Status  Logged Active Pages SPg/s MDC/s Share
 SLSSP201  14.1 8.434 7.280  1.16   210  210   .0     .0   .0   .0 EME,CL0,DISP      1      1     0   .00    .0   100
```

```
 Data  <--------- Paging Activity/s ----------> <--------------- Number of Pages ---------------->
         Spaces <Page Rate>   Page <--Page Migration-->               <-Resident-> <--Locked-->          Stor
 Userid   Owned Reads Write Steals  >2GB> X>MS MS>X X>DS   WSS Resrvd  R<2GB  R>2GB L<2GB L>2GB  XSTOR  DASD Size
 SLSSP201     0   .0    .0    .0      .0   .0   .0   .0 261612      0  53814 207810     2    10      0     1 1024M
```

```
sm slssp201 app susp
Ready; T=0.01/0.01 13:44:43
SLSSP201: PM: Syncing filesystems ... done.
SLSSP201: Freezing user space processes ... (elapsed 0.01 seconds) done.
SLSSP201: Freezing remaining freezable tasks ... (elapsed 0.01 seconds) done.
SLSSP201: PM: Preallocating image memory... done (allocated 23314 pages)
SLSSP201: PM: Allocated 93256 kbytes in 0.02 seconds (4662.80 MB/s)
SLSSP201: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
SLSSP201: HCPGSP2627I The virtual machine is placed in CP mode due to a SIGP initial CPU reset from CPU 00.
SLSSP201: PM: freeze of devices complete after 19.501 msecs
SLSSP201: PM: late freeze of devices complete after 0.004 msecs
SLSSP201: PM: Wrote 92860 kbytes in 0.47 seconds (197.57 MB/s)
SLSSP201: PM: S|
SLSSP201: md: stopping all md devices.
SLSSP201: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.

SLSSP201: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
```

```
 Data  <--------- Paging Activity/s ----------> <--------------- Number of Pages ---------------->
         Spaces <Page Rate>   Page <--Page Migration-->               <-Resident-> <--Locked-->          Stor
 Userid   Owned Reads Write Steals  >2GB> X>MS MS>X X>DS   WSS Resrvd  R<2GB  R>2GB L<2GB L>2GB  XSTOR  DASD Size
 SLSSP201     0   .0  19.1 1134.7    .0   .0   .0   .0 261631      0      0      0     0     0      0 24722 1024M
```

Use on CDSL

- Enroll/remove user in reboot server
- Remove user from allocation database
- Add/remove NAT for a user
- Query allocation database

Reboot Server

```
ACTION=="change", ENV{SMSG_TEXT}=="ADDUSR*", \
   PROGRAM=="/sbin/newruser $env{SMSG_TEXT}", \
   RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
cat /sbin/newruser
#!/bin/sh
/sbin/chccwdev -e 01ff
sleep 1
/usr/sbin/useradd -m -s /sbin/reboot.sh $2
/bin/mkdir /home/$2/.ssh
/sbin/cmsfscat  -d /dev/disk/by-path/ccw-0.0.01ff $2.KEY > /home/$2/.ssh/authorized keys
/sbin/cmsfscat  -d /dev/disk/by-path/ccw-0.0.01ff $2.KEY > /home/$2/.ssh/authorized keys2
sleep 1
/sbin/chccwdev -d 01ff

echo "Finished ADDUSR"
```

SHARE
in Anaheim
2012

Web Server

```
ACTION=="change", ENV{SMSG_TEXT}=="QUERY*", \
  PROGRAM=="/sbin/lcdsquery $env{SMSG_TEXT}", \
  RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
cat /sbin/lcdsquery
#!/bin/sh
/usr/bin/mysql --user=xxxx --password=yyyy <<STOP
connect lcds;
select * from lcds_user where userid = "$2";
\q
STOP
```

```
sm lcdsweb app query l003002
Ready; T=0.01/0.01 20:42:54
 20:42:54  * MSG FROM LCDSWEB : USERID IPADDR STATE L003002 63.90.228.1 ALLO
```

Web Server

```
ACTION=="change", ENV{SMSG_TEXT}=="FREE*", \
  PROGRAM=="/sbin/firstfree $env{SMSG_TEXT}", \
  RUN+="/sbin/vmcp msg $env{SMSG_SENDER} '$result'"
```

```
cat /sbin/firstfree
#!/bin/sh
mysql --user=xxxx --password=yyyy <<STOP
connect lcds;
select * from lcds_user where state = 'FREE' order by userid limit 1;
\q
STOP
```

```
sm lcdsweb app free
Ready; T=0.01/0.01 16:08:34

 16:08:35  * MSG FROM LCDSWEB : USERID IPADDR STATE L004068 63.90.228.83 FREE
```

Summary

- smsgiucv_app receives SMSGs from other virtual machines and presents them to userspace as udev change events

- udev rules provide great flexibility for executing commands and programs

- No network connection needed between source and target virtual machine

- Works across members in an SSI cluster

SHARE
in Anaheim
2012