開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
⠠⠠⠠⠠
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Light introduction to process management

Miroslav Osladil

Agenda

- Introduction
- Basic division
- SysV, BSD, GNU, POSIX
- List of some interesting commands
- Demo

Basic division

- Standard toolset is SysV
- BSD toolset
- POSIX standards
- GNU tools
- All other user compiled

SUN world

- Why don't use GNU variant only?
  Binary Compatibility Guarantee
- Why several versions of same tools?
  SUN follows standards

Many programs and systems could expect the same input and output as well.

SUN world

- SysV binary tools in /usr/bin
- BSD binary tools in /usr/ucb

Example: ls(1)

```
# find / -name ls
/usr/bin/ls
/usr/bin/amd64/ls
/usr/ucb/ls
/usr/xpg6/bin/ls
/usr/xpg4/bin/ls
/usr/gnu/bin/ls
```

GNU world

Our customer is our master, SUN says.

- GNU toolset in /usr/gnu/bin
- Sun Freeware in /usr/sfw/bin

Sun Freeware available on SUN website or on SUN Companion CD's

If you want use one toolset instead of another one update your PATH enviroment.

List of some interesting commands

- apptrace – for trace of library calls
- dtrace – debugger, new in version 10
- pargs – get list of arguments and environment variables with which process was started
- pfiles – list of file descriptors, associated with process
- pgrep – get the PID's of processes by name i.e. Something like ps -efl|grep -v grep|grep process_name
- pkill – send signal to process. For example pkill -9 init :-P
- pldd – list dynamic libraries, associated with process, similar to ldd for executable
- plockstat – see list of locked by process files. Lock can be mutex i.e. exclusive and reader/writer for shared access
- pmap – get memory map (segments) of process
- preap – try to kick-off zombie process
- prstat – fullscreen view of processes sorted by different criteria, similar to Linux top command
- prun – continue hold with pstop process
- ps - print process information and status. In Solaris exist SYSV and BSD variants, respectively /usr/bin/ps and /usr/ucb/ps
- psig – list signals that can be handled by process
- pstack – get backtrace stack of process for debugging purposes
- pstop – temporary hold process
- ptree - print the tree of processes
- pwait – wait till process finish
- pwdx – list working directory for process, like pwd command
- truss – for trace system calls and signals

Let's play

```
for x in $(ls /usr/bin);  do
    man $x
done
```

pargs(1)

Get list of arguments and environment variables with which process was started

```
# ps -ef|grep -v grep|grep Xorg
    mira  2410  2274   1   Nov 03 ?          54:57 /usr/X11/bin/Xorg :0 -depth 24 -nolisten
     tcp -nobanner -auth /var/dt/A:0-e3aqCe
# pargs 2410
2410:   /usr/X11/bin/Xorg :0 -depth 24 -nolisten tcp -nobanner -auth /var/dt/A:0-e3aqCe
argv[0]: /usr/X11/bin/Xorg
argv[1]: :0
argv[2]: -depth
argv[3]: 24
argv[4]: -nolisten
argv[5]: tcp
argv[6]: -nobanner
argv[7]: -auth
argv[8]: /var/dt/A:0-e3aqCe
```

pfiles(1)

List of file descriptors associated with PID

```
# pfiles 428
---cut---
  57: S_IFREG mode:0644 dev:182,65545 ino:2102 uid:101 gid:10 size:17408
      O_RDWR|O_CREAT|O_LARGEFILE FD_CLOEXEC
      /export/home/mira/.mozilla/firefox/xypbu2c2.default/downloads.sqlite
---cut---
     5: S_IFSOCK mode:0666 dev:302,0 ino:27225 uid:0 gid:0 size:0
      O_RDWR
SOCK_DGRAM
SO_REUSEADDR,SO_SNDBUF(57344),SO_RCVBUF(57344)
sockname: AF_INET 10.109.232.135  port: 68
   ---cut---
```

pgrep(1), pkill(1)

- pgrep(1)
  - Something like is Debian's 'pidof named'
  - ps -efl|grep -v grep|grep process_name|awk '{print $1}'

```
# pgrep named
586
# ps -e|grep -v grep|grep named|awk '{print $1}'
586
```

- pkill(1)

Instead of going through all PIDs you want to kill, you can simply use the pkill command if you already know the process name or part of it.

```
# pkill firefox
```

ps(1), ptree(1)

As said before there is more than one ps(1) binary

- /usr/bin/ps -ef
- /usr/ucb/ps aux

But, there is no such 'ps afx'...

- /usr/bin/ptree

prstat(1)

The most often question when running opensolaris for the first time is...

Where is my top(1)?

The answer is simple, use prstat(1) instead. It does the same as top does plus much more. See switches. You can monitor processes per user, processor, zone ....

```
# prstat -U 101
```

prun(1), pstop(1)

- You can pause and run existing PID
- Unfortunately there is no so much reason of usage in daily tasks

```
# pstop $(pgrep sshd)
# ssh localhost
(Press Ctrl-C after long long time of waiting)
^C
# prun $(pgrep sshd)
# ssh localhost
Password:
Last login: Tue Nov 16 23:04:37 2008 from 10.109.232.135
Sun Microsystems Inc.   SunOS 5.11      snv_101a November 2008
#
```

pwdx(1)

Display work directory of the process.

It can be useful when unmounting filesystem or just need to know what place process was called.

Any way, fuser(1) does much more

```
# pwdx 24194
24194:  /rpool/zone1/root/export/home/user
```

apptrace(1), truss(1)

If you want to know what your app does

apptrace

- trace application function calls to shared libraries

truss

- trace system calls and signals, arguments, memory...

開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
⠠⠕⠏⠑⠝
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Thank you!

Miroslav Osladil
mira@osladil.cz

http://www.opensolaris.org/os/project/czosug/

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan