System & Service Management

Solaris Commands Guide

System Commands	
Show general system information	`# prtdiag [-v]`
Change Runlevel (6 = Restart)	`# init 6`

File System Commands	
Creating New Files[1]	`$ touch NewFilename`
Creating New Directories	`$ mkdir [-Options] NewDirectory` Options: -p create the whole path provided
Listing Files	`$ ls`
Copying Files or Directories	`$ copy [-Options] OldFilename NewFilename` Options: -i = no overwrite -r = recursive copy (incl subdirs)
Copying Multiple Files	`$ copy [-Options] File1 File2 TargetDir`
Moving or Renaming Files	`$ mv [-Options] OldFilename NewFilename` Options:
Deleting Files	`$ rm [-Options] FileToDelete` Options: -i ask before delete -r deletes subcontent * wildcard for FilesToDelete
Deleting Empty Directories	`$ rmdir [-Options]DirectoryToDelete`
Displaying File Content	`$ cat` more
Displaying File Type	`$ file *`
Create a Symbolic Link File	`$ ln -s SourceFile TargetFile`
Changing Permissions[2]	`# chmod 777`
Search for regular expression (globally)	`$ grep searchstring /etc/group /etc/…`
Count number of word occurrance	`$ grep -i searchword /etc/system`
Find and locate files and directories	`$ find /etc -name "*exe"`
Find all files that were modified the day before	`$ find /export/home/user -mtime -1`

Processors and Cores	
Show CPU information	`# psrinfo [-v]`

Processes and Threads (LWP's)	
Show Process Status of all Processes	`# ps -ef`
Search for a certain Process	`# ps -ef | grep prime`
Examine active Processes	`# prstat`
Report Per-Processor Statistics	`# mpstat` *`[interval]`* CPU processor ID

[1] http://docs.sun.com/app/docs/doc/806-7612/6jgfmsvpo?l=en&a=view

[2] http://en.wikipedia.org/wiki/Chmod

	minf minor faults mjf major faults xcal inter-processor cross-calls intr interrupts ithr interrupts as threads (not counting clock interrupt) csw context switches icsw involuntary context switches migr thread migrations (to another processor) smtx spins on mutexes (lock not acquired on first try) srw spins on readers/writer locks (lock not acquired on first try) syscl system calls usr percent user time sys percent system time wt percent wait time idl percent idle time
Determine a Process. Default signalID is 15. Use signalID 9 as required.	# kill <signalID> <ProcessID> (See Show All Processes to get ProcessID)
Determine a Process (Alternative Method)	# pkill dtmail
Change a job from background to foreground	# fg %<JobID>
List the current running jobs	# bg
Change a job from foreground to background	# bg %<JobID>

SUN Storage Manager

List Disk Partitions	# format Searching for disks...done AVAILABLE DISK SELECTIONS: 0. c1t0d0 <DEFAULT cyl 4092 alt 2 hd 128 sec 32> /pci@0,0/pci15ad,1976@10/sd@0,0 1. c1t1d0 <DEFAULT cyl 1021 alt 2 hd 128 sec 32> /pci@0,0/pci15ad,1976@10/sd@1,0 2. c1t2d0 <DEFAULT cyl 1021 alt 2 hd 128 sec 32> /pci@0,0/pci15ad,1976@10/sd@2,0
Create metadatabase for redundant storage of logical disk information. -c 2 = create two copies of metadb -f = force first disk	# metadb -a -c 2 -f c1t1d0s7 # metadb -a -c 2 c1t2d0s7
Show metadb information	root@s0011:~# metadb

```
        flags           first blk       block count
     a        u         16              8192            /dev/dsk/c1t1d0s7
     a        u         8208            8192            /dev/dsk/c1t1d0s7
     a        u         16              8192            /dev/dsk/c1t2d0s7
     a        u         8208            8192            /dev/dsk/c1t2d0s7
```

Construct a new File System	root@s0011:~# newfs /dev/rdsk/c1t1d0s0 newfs: /dev/rdsk/c1t1d0s0: (y/n)? y
Mount a File System	mount /dev/dsk/c1t1d0s0 /mnt
Unmount a File System	umount /mnt
Show Capacity Information of Mount Point /mnt	df -h /mnt
Create RAID0 with 2 Disks	# metainit -f d0 2 1 c1t1d0s0 1 c1t2d0s0
Extend an existing File System	growfs -M /mnt /dev/md/rdsk/d0

Create a RAID1 with 2 Disks	root@s0011:~# metainit d0 1 1 c1t1d0s0 d0: Concat/Stripe is setup root@s0011:~# metainit d1 1 1 c1t2d0s0 d1: Concat/Stripe is setup root@s0011:~# metainit d5 -m d0 d5: Mirror is setup root@s0011:~# metattach d5 d1 d5: submirror d1 is attached root@s0011:~# newfs /dev/md/rdsk/d5 root@s0011:~# mount /dev/md/dsk/d5 /mnt
Destroy an existing RAID array	# metaclear d5 # metaclear d1 # metaclear d0
Remove State Databases	# metadb -d -f c1t1d0s7 # metadb -d -f c1t2d0s7

ZFS Volume Management

Create new ZFS Pool with Disk1	# zpool create -f my_pool c1t1d0
Create a RAID1 set by attaching a second disk	# zpool attach -f my_pool c1t1d0 c1t2d0
Detach Disk from Pool	# zpool detach my_pool c1t2d0
Show ZFS Pool	# zpool status

```
pool: my_pool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Fri Nov  5 10:37:54 2010
config:

        NAME        STATE     READ WRITE CKSUM
        my_pool     ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0  73.5K resilvered

errors: No known data errors
```

Get verbose Information about Pool	# zfs get all my_pool
Show Capacity Information of Pool	# df -h /my_pool

```
Filesystem             size   used  avail capacity  Mounted on
my_pool                3.9G    21K   3.9G     1%    /my_pool
```

Create File Systems	# zfs create my_pool/home # zfs create my_pool/home/my_home # zfs create my_pool/home/your_home # zfs create my_pool/home/his_home # zfs create my_pool/home/her_home # zfs create my_pool/home/our_home
List all Pools	# zfs list

```
NAME                    USED  AVAIL  REFER  MOUNTPOINT
my_pool                 428K  3.91G    23K  /my_pool
my_pool/home            132K  3.91G    27K  /my_pool/home
my_pool/home/her_home    21K  3.91G    21K  /my_pool/home/her_home
my_pool/home/his_home    21K  3.91G    21K  /my_pool/home/his_home
my_pool/home/my_home     21K  3.91G    21K  /my_pool/home/my_home
my_pool/home/our_home    21K  3.91G    21K  /my_pool/home/our_home
my_pool/home/your_home   21K  3.91G    21K  /my_pool/home/your_home
```

Define Quota for a File System (Use `zfs list` to check quota limitations)	# zfs set quota=50m my_pool/home
Set File System Reservation	# zfs set reservation=15m my_pool/home/my_home

Set Mountpoint for File System (Use `ls /export/home/` to validate this command)	`# zfs set mountpoint=/export/home my_pool/home`
Remove a Mountpoint	`# zfs umount -f /export/zones`
Remove particular ZFS Pools (-r for recursive)	`# zfs destroy -r my_pool/home`
Remove particular ZPool	`# zpool destroy my_pool`

File Archiving and Compression

Create a new Tar Archive	`# tar cvf archive.tar file1 file2 file3`
List Table of Contents of Tar File	`# tar tf archive.tar`
Extract Files from Tar File	`# tar xvf archive.tar`
Create a new (compressed) Jar Archive	`# jar cvf archive.jar`
Compress a single file. (Creates a .Z file archive)	`# compress -v archive.tar`
View compressed files	`# zcat file1.z | more`
Uncompress files	`# uncompress -v file1.tar.Z`
Compress a single file using 7z	`# 7za a file1.7z file1`
Extract files from a 7z archive	`# 7za x file1.7z`
Compress a single file using Gzip	`# gzip file1`
Uncompress files using Gunzip	`# gunzip file1.gz`
View Gzip compressed files	`# gzcat file1.gz`
Compress files using Zip	`# zip file.zip file1 file2 file3`
Uncompress Zip/Jar files using Unzip	`# unzip file.zip`
View Zip compressed files	`# unzip -l file.zip`

Remote Sessions

Remote Login using rlogin	`$ rlogin -l username hostname`
Run a Program on a Remote System	`$ rsh hostname command`
Remote Login using telnet	`$ telnet hostname`

Remote Copy Command	`$ rcp sourcefile hostname:destfile`
	`$ rcp hostname:sourcefile destfile`
	`$ rcp hostname:sourcef hostname:destf`

Network Management

Show Network Devices	`# dladm show-dev`
Disable Network Device	`# ifconfig e1000g1 plumb`
Set IP Configuration	`# ifconfig e1000g1 172.16.0.1 netmask 255.255.255.0 up`
Show Routing Table	`# netstat -rn`

Virtual Zones

Create new Virtual Zone	`# zonecfg -z z0100`

	`Zonecfg:zonename> create`
	`Zonecfg:zonename> set zonepath=/export/zones/z0100`
	`Zonecfg:zonename> set autoboot=true`
	`Zonecfg:zonename> add net`
	`Zonecfg:zonename:net> set address=z0100`
	`Zonecfg:zonename:net> set physical=e1000g1`
	`Zonecfg:zonename:net> end`
	`Zonecfg:zonename> info`
	`Zonecfg:zonename> verify`
	`Zonecfg:zonename> commit`
	`Zonecfg:zonename> exit`

Install new Virtual Machine	`# zoneadm -z z0100 install`
List Virtual Zones	`# zoneadm list -cv`

```
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   1 z0100            running    /export/zones/z0100            native   shared
```

Start up a Virtual Zone	`# zoneadm -z z0100 boot`
Logon to Virtual Zone	`# zlogin -e ^ -C z0100`

Logout of a Virtual Zone	`# exit` `# exit`
Change from Virtual Zone to Host OS	`# ^.`
Show Processes Summary of a Zone[3]	`# prstat -Z`

Fair Share Scheduler

Start Fair Share Scheduler	`# dispadmin -d FSS`
	`# priocntl -s -c FSS -i class TS`
	`# priocntl -s -c FSS -i pid 1`

Assign CPU Shares to Global Zone	`prctl -n zone.cpu-shares -r -v 10 -i zone global`
Assign CPU Shares to Virtual Zones	`# prctl -n zone.cpu-shares -r -v 10 -i zone z0100` `# prctl -n zone.cpu-shares -r -v 1 -i zone z0101`

Write CPU Shares (Make them effective after Restart)	`global # zonecfg -z z0100`
	`zonecfg:z0100> add rctl`
	`zonecfg:z0100> set name=zone.cpu-shares`
	`zonecfg:z0100:rctl> add value (priv=privileged,limit=10,action=none)`
	`zonecfg:z0100:rctl> end`
	`zonecfg:z0100> exit`

[3] http://www.solarisinternals.com/wiki/index.php/Processes

vi Editor

vi has two different modes: Command Mode and Text Entry Mode:

Command Mode	In Command Mode everything that you type is interpreted as a command. For most commands, a single keystroke is the entire command, while other commands require more information, such as a linecount followed by a single-character command. Command mode is used to perform powerful editing functions, as well as such actions as writing the text buffer to a file. To enter Command Mode from Edit Mode, type an <ESC> character.
Edit Mode	In Edit Mode all characters typed are printed on the screen and become part of the current document. Editing in this mode is restricted to insertion and typeover. To enter Edit Mode from Command Mode, type the character **'i' (for insert mode), 'a' (for append mode),** or **'o' (for open mode)**.

Command	Function
a	Appends text after the cursor
A	Appends text at the end of the line
i	Inserts text before the cursor
I	Inserts text at the beginning of the line
o	Opens a new line below the cursor
O	Opens a new line above the cursor
:r *filename*	Inserts text from another file into the current file

Key Sequence	Cursor Movement
h, left arrow, or Backspace	Left one character
j or down arrow	Down one line
k or up arrow	Up one line
l, right arrow, or space bar	Right (forward) one character
b	Back one word
w	Forward one word
e	To the end of the current word
$	To the end of the line
0 (zero)	To the beginning of the line
^	To the first non-white space character on the line
Return	Down to the beginning of the next line

Last Line Mode	Enter a colo (:) to change from Command Mode into Last Line Mode. This is used to enter advanced commands.

Open a File	`$ vi [-Options] Filename`
	`Options:`
	`-R Open File in Readonly Mode`

View a File (Readonly)	`$ view Filename`
Save a File	`:w`
Save and Quit	`:wq`
Quit without Saving	`:q!`

Further Information

http://en.wikipedia.org/wiki/Fork_bomb
http://en.wikipedia.org/wiki/Vi
http://www.sun.com/blueprints/0505/819-2679.pdf