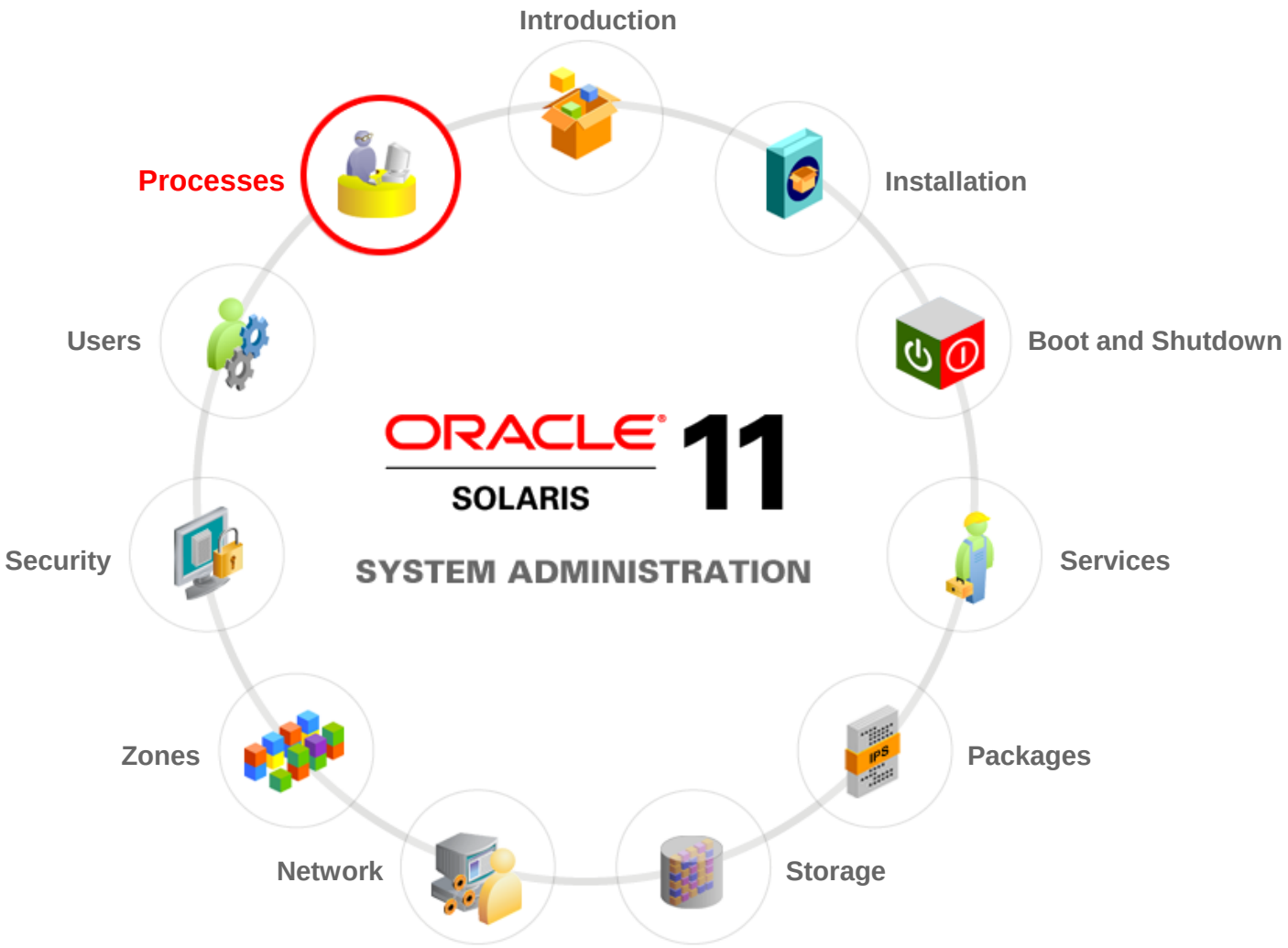
11

Managing System Processes and Scheduling System Tasks

Workflow Orientation



Objectives

After completing this lesson, you should be able to:

- Explain system processes management
- Manage system processes
- Schedule system administration tasks

Agenda

- **Managing System Processes**
- Scheduling System Administration Tasks

Importance of System Processes Management

System processes management ensures that you can:

- Determine what processes are running in the system
- Determine what state a process is in
- Determine which processes are using the greatest percentage of system resources
- Control processes
- Terminate unwanted processes
- Schedule routine tasks

System Processes: Overview

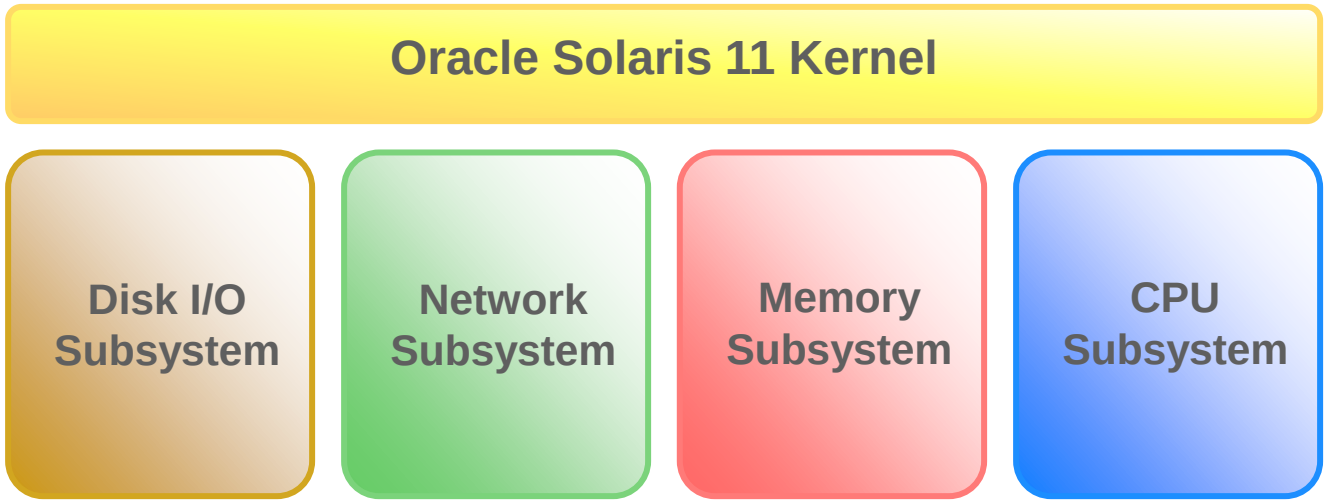
A process is:

- Any program that is running in the system
- Assigned a unique process identification (PID) number that is:
  - Used by the kernel to track, control, and manage a process
  - Displayed by using the `ps` or `pgrep` command

Parent and Child Processes

- When one process creates another:
  - The first process is considered the parent process, which is identified by a parent process ID (PPID) number
  - The new process is called the child process
- The parent and child processes interact as follows:
  - While the child process runs, the parent process waits.
  - When the child process finishes its task, it informs the parent process.
  - The parent process then terminates the child process.
  - If the parent process is an interactive shell, a prompt appears, indicating that it is ready for a new command.

Identifying the Process Subsystems



Identifying the Process States

A process can be in one of the following states:

State	Description
`run`	The process is in the run queue and running on a CPU.
`sleep`	The process is waiting for work.
`zombie`	The parent process has terminated.
`stop`	The process is stopped.

Commands for Managing Processes

Command	Description
`ptree`	Displays the process trees for the specified process ID
`ps`	Displays detailed information about the active processes in the system
`pgrep`	Displays information about a process based on specific criteria
`prstat`	Displays statistics for the active processes in a system
`pstop`	Stops each process
`prun`	Starts each process

Terminating Unwanted Processes

- Users can terminate any process that they own.
- Users with the `root` role can kill any process in the system.
- Two commands are used to terminate processes: `kill` and `pkill`.

Signal Number	Signal Name	Event	Default Action
1	SIGHUP	Hangup	Exit
2	SIGINT	Interrupt	Exit
9	SIGKILL	Kill	Exit
15	SIGTERM	Terminate	Exit

Managing System Processes

- Viewing the parent/child process relationship
- Listing system processes
- Displaying information about processes
- Displaying active process statistics
- Stopping and starting a system process
- Killing a process

Viewing the Parent/Child Process Relationship

To view the parent/child process relationship, use `ptree pid`.

```
# ps -ef
   UID   PID  PPID   C    STIME TTY          TIME CMD
---
---
oracle   1345 1280   0   Jul 31 ?          0:01 gnome-panel
---
---

# ptree 1345
1032   /usr/sbin/gdm-binary
  1046   /usr/lib/gdm-simple-slave --display-id
    /org/gnome/DisplayManager/Displa
    1258   /usr/lib/gdm-session-worker
      1280   gnome-session
        1345   gnome-panel
```

Listing System Processes

To list the active processes in a system, use `ps`.

```
# ps
  PID TTY          TIME CMD
 4605 pts/4        0:00 bash
 4604 pts/4        0:00 su
 5880 pts/4        0:00 ps
```

Listing System Processes

To generate a full listing of every process that is currently running, use `ps -ef`.

```
# ps -ef
  UID   PID  PPID   C    STIME TTY          TIME CMD
  root    0     0   0 06:50:42 ?           0:02 sched
  root    5     0   0 06:50:40 ?           0:02 zpool-rpool
  root    6     0   0 06:50:40 ?           0:02 kmem_task
  root    1     0   0 06:50:43 ?           0:00 usr/sbin/init
...
...
...
```

Displaying Information About Processes

To display the PID of a particular process, use `pgrep` *`process`*.

```
# pgrep sched
0
9179
29414
```

```
# pgrep -l manager
  4238 updatemanagerno
  4283 nwam-manager
```

Displaying Active Process Statistics

To display statistical information about running processes, use `prstat`.

```
# prstat
   PID USERNAME  SIZE   RSS STATE    PRI NICE      TIME  CPU PROCESS/NLWP
------------------------------------------------------------------------------
 26264 root       38M  372K run      10    0 183:40:15  95% sysconfig/1
  4297 oracle     99M   75M run      49    0   2:33:14 0.8% java/20
   739 root       39M 9552K sleep    59    0   2:14:44 0.6% pkg.depotd/64
  4668 oracle    131M   20M sleep    59    0   0:01:40 0.6% gnome-terminal/2
   832 oracle     73M   46M sleep    59    0   0:04:55 0.5% Xorg/3
  5890 root       11M 3244K cpu0     49    0   0:00:00 0.3% prstat/1
   519 root       19M 6212K sleep    59    0   0:09:17 0.1% named/4
  4185 oracle    128M   16M sleep    59    0   0:00:07 0.0% metacity/1
  4605 root       10M 2672K run      39    0   0:00:00 0.0% bash/1
  4289 oracle    134M   19M sleep    59    0   0:04:33 0.0% isapython2.6/1
  7605 root       14M 4408K sleep    59    0   0:02:40 0.0% nscd/120
    15 root       20M   16M sleep    59    0   0:04:55 0.0% svc.configd/27
  4238 oracle     62M   27M sleep    12   19   0:04:43 0.0% updatemanagerno/1
Total: 198 processes, 1075 lwps, load averages: 1.42, 1.39, 1.43
```

Displaying Active Process Statistics

```
# prstat -s cpu 20 3
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
------------------------------------------------------------------------
 26264 root       38M  372K run      30    0 186:38:44  96% sysconfig/1
  4297 oracle     99M   75M sleep    49    0   2:34:36 0.8% java/20
   739 root       39M 9552K run      59    0   2:15:45 0.6% pkg.depotd/64
  4668 oracle    131M   20M sleep    59    0   0:01:41 0.2% gnome-terminal/2
  5987 root       11M 3620K cpu0     59    0   0:00:00 0.2% prstat/1
<output omitted>
Total: 199 processes, 1078 lwps, load averages: 1.45, 1.40, 1.38
```

```
# prstat -s rss 20 3
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
------------------------------------------------------------------------
  4297 oracle     99M   75M run      39    0   2:34:38 0.8% java/20
   528 root       61M   58M sleep    59    0   0:00:52 0.0% hald-addon-acpi/1
   832 oracle     74M   47M sleep    59    0   0:05:00 0.3% Xorg/3
 26129 oracle    142M   43M sleep    49    0   0:01:13 0.0% nautilus/3
  4210 oracle    147M   31M sleep    49    0   0:00:04 0.0% nautilus/1
  1354 root      141M   29M sleep    59    0   0:00:03 0.0% gedit/1
  4238 oracle     62M   27M run      12   19   0:04:46 0.0% updatemanagerno/1
  5894 oracle    138M   25M sleep    49    0   0:00:01 0.0% gedit/1
<output omitted>
Total: 199 processes, 1077 lwps, load averages: 1.38, 1.38, 1.37
```

Stopping and Starting a System Process

1. Using `pgrep` *`process`*, obtain the process ID of the process that you want to control.
2. Temporarily stop the process by using `pstop` *`pid`*.
3. Verify that the process has stopped by using `ps -ef | grep` *`pid`*.
4. Restart the process by using `prun` *`pid`*.
5. Verify that the process has restarted by using `ps -ef | grep` *`pid`*.

Stopping and Starting a System Process: Example

```
# pgrep rptpgm
3366
# pstop 3366
# ps -ef | grep 3366
root  3366  2864  47 16:09:54 pts/2   0:48 dd if=/dev/zero
of=/dev/null
# ps -ef | grep 3366
root  3366  2864  47 16:09:54 pts/2   0:48 dd if=/dev/zero
of=/dev/null

# prun 3366
# ps -ef | grep 3366
root  3366  2864  47 16:10:17 pts/2   0:52 dd if=/dev/zero
of=/dev/null
# ps -ef | grep 3366
root  3366  2864  47 16:10:20 pts/2   1:01 dd if=/dev/zero
of=/dev/null
```

Killing a Process

1. Obtain the process ID of the process that you want to terminate by using `pgrep` *`process`*.
2. Terminate the process by using `kill [-signal] pid` or `pkill [-signal]` *`process`*.
3. Verify that the process has been terminated by using `pgrep pid` or `pgrep` *`process`*.

```
$ pgrep -l mail
215 sendmail
470 dtmail
$ pkill dtmail
$ pgrep -l mail
215 sendmail
$
```

Process Management Commands: Summary

Command	Description
`ps`	Displays information about the active processes in a system
`pgrep`	Displays information about a process based on specific criteria
`prstat`	Displays statistics for the active processes in a system
`kill, pkill`	Terminates a process

Quiz

What state is a parent process in when it is waiting for an event to complete?

a. run
b. sleep
c. zombie
d. stop

Quiz

What state is a parent process in when it is waiting for an event to complete?

a. run

b. sleep

c. zombie

d. stop

Quiz

When used with `kill` or `pkill`, which signal terminates a process instantly with no opportunity to perform an orderly shutdown?

a. 1, SIGHUP

b. 2, SIGINT

c. 9, SIGKILL

d. 15, SIGTERM

Quiz

When used with `kill` or `pkill`, which signal terminates a process instantly with no opportunity to perform an orderly shutdown?

a. 1, SIGHUP

b. 2, SIGINT

c. 9, SIGKILL

d. 15, SIGTERM

Agenda

- Managing System Processes
- **Scheduling System Administration Tasks**

Scheduling a Single Job Using the `at` Command

- You can schedule a job for execution at a later time by using the `at` command.
- The job can consist of a single command or a script.
- The `at` command allows you to schedule the automatic execution of routine tasks.
- `at` files execute their tasks once after which they are removed from their directory.
- The `at` command is most useful for running simple commands or scripts that direct output into separate files for later examination.

Creating an `at` Job

1. Start the `at` utility, specifying the time you want your job to be executed.
2. At the `at` prompt, type the commands or scripts that you want to execute, one per line.
3. Press Control-D to exit the `at` utility and save the `at` job.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> <Press Control-D>
job 897355800.a at Thu Jul 12 19:30:00 2015
```

at Commands

Command	Description
atq	Displays status information about the at jobs that you have created **Note:** You can also use this command to verify that you have created an at job.
at -l *[job-id]*	Displays information about the execution times of your at jobs
at -r *[job-id]*	Removes the at job from the queue before the job is executed

Denying Access to the `at` Command

1. Assume the `root` role.
2. Edit the `/etc/cron.d/at.deny` file by using the `pfedit` command.
3. Add the names of users, one username per line, that you want to prevent from using the `at` commands.

```
$ pfedit /etc/cron.d/at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
...
```

Scheduling Repetitive System Tasks

- Repetitive tasks can be:
  - Executed automatically by using the `cron` facility
  - Scheduled to run daily, weekly, or monthly
- The `cron` facility:
  - Uses `crontab` files for scheduling and maintaining routine tasks
  - Is controlled by the clock daemon, `cron`
- The `cron` daemon:
  - Checks for new `crontab` files
  - Reads the execution times that are listed within the files
  - Submits the commands for execution at proper times
  - Listens for notifications from the `crontab` commands about updated `crontab` files

Interpreting the `crontab` File Format

```
10 3 * * 0 /usr/sbin/logadm
```

Field	Range of Values
minute	0 to 59; * means every minute.
hour	0 to 23; * means every hour.
day of month	1 to 31; * means every day of the month.
month	1 to 12; * means every month.
day of week	0 to 6; * means every day of the week. Sunday is 0.
command	This is the full path name to the command to be run.

Displaying the Default `root cron` File

```
# crontab -l
#ident "%Z%%M% %I%   %E% SMI"
<header and copyright content omitted>
#
# The root crontab should be used to perform accounting data
collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] &&
/usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

crontab Files

- The files are maintained in /var/spool/cron/crontabs.
- Access to the files is controlled through:
  - /etc/cron.d/cron.allow
  - /etc/cron.d/cron.deny
- Only specified users are permitted to perform crontab tasks based on the access files, as follows:
  - If the cron.allow file exists, only the users listed in this file can create, edit, display, or remove the crontab files.
  - If the cron.allow file does not exist, all users, except the users listed in the cron.deny file, can create, edit, display, or remove the crontab files.
  - If neither file exists, only the user with the root role can run the crontab command.

The Default `cron.deny` File

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
```

Scheduling System Administration Tasks

- Scheduling repetitive system tasks
- Administering `crontab` files

Scheduling Repetitive System Tasks

1. Set up `vi` as the default editor by using `EDITOR=vi`.
2. Create a new `crontab` file by using `crontab -e [username]`.
3. Verify that your `crontab` file changes by using `crontab -l [username]`.
4. Verify that the `crontab` file exists by using `ls -l /var/spool/cron/crontabs`.

Scheduling Repetitive System Tasks: Example

```
# EDITOR=vi
# export EDITOR
# crontab -e jjones
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
:wq
# crontab -l jjones
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
# ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Sep 19 16:23 adm
-rw-------  1 root     staff        225 Nov  5 09:19 jjones
-rw-r--r--  1 root     root        1063 Nov  5 16:23 lp
-rw-r--r--  1 root     sys          441 Sep 19 16:25 root
-rw-------  1 root     staff         60 Nov  5 09:15 smith
-rw-r--r--  1 root     sys          308 Sep 19 16:23 sys
```

Administering `crontab` Files

- Removing a `crontab` file
- Denying `crontab` command access
- Limiting `crontab` command access to specified users

Removing a `crontab` File

To remove a `crontab` file, use `crontab -r` *username*.

```
# crontab -r jjones
```

To verify that the `crontab` file has been removed, use `ls -l /var/spool/cron/crontabs`.

```
# ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Sep 19  16:23 adm
-rw-r--r--  1 root     root        1063 Nov  5  16:23 lp
-rw-r--r--  1 root     sys          441 Sep 19  16:25 root
-rw-------  1 root     staff         60 Nov  5  09:15 smith
-rw-r--r--  1 root     sys          308 Nov 19  16:23 sys
```

Denying `crontab` Command Access

1. Change directories to `/etc/cron.d`.
2. Using the `vi` text editor, add an entry to the `cron.deny` file for each user.
3. Verify that the users are listed in the file.

```
# cd /etc/cron.d
/etc/cron.d# vi cron.deny
daemon
bin
smtp
nuucp
jjones
/etc/cron.d# grep jjones cron.deny
jjones
```

Limiting `crontab` Access to Specified Users

1. Change directories to `/etc/cron.d`.
2. Using the `vi` text editor, create the `cron.allow` file and add an entry for each additional user.
3. Verify that `root` and the other users are listed in the file by using `cat cron.allow`.

```
# cd /etc/cron.d
/etc/cron.d# vi cron.allow
omai
jsmith
tbone
/etc/cron.d# cat cron.allow
omai
jsmith
tbone
```

Quiz

If the `cron.allow` file does not exist, all users (except the users listed in the `cron.deny` file) can create, edit, display, or remove the `crontab` files.

a. True
b. False

Summary

In this lesson, you should have learned how to:

- Explain system processes management
- Manage system processes
- Schedule system administration tasks

Practice 11: Overview

- 11-1: Managing System Processes
- 11-2: Scheduling System Tasks