9

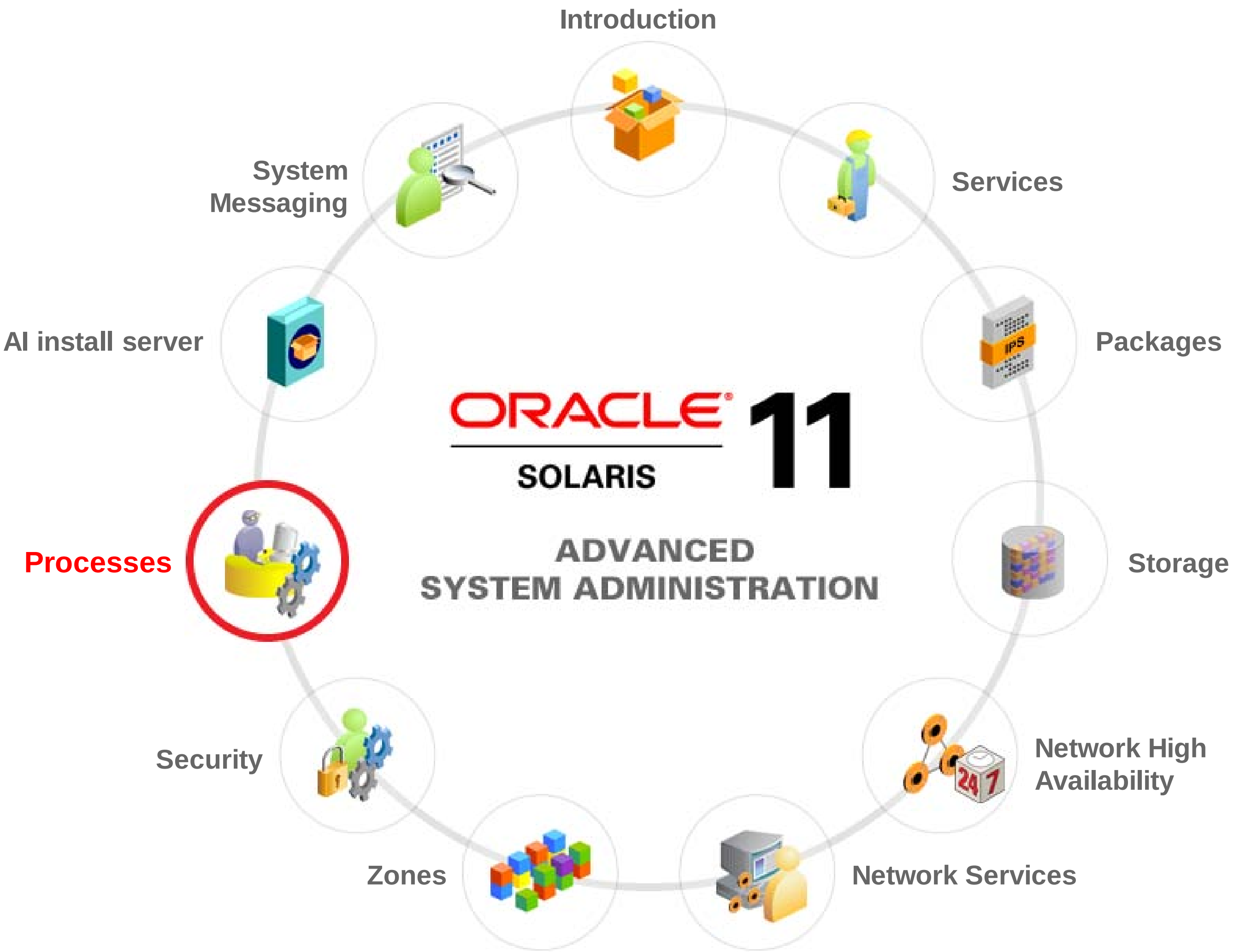
Managing Processes and Priorities

Objectives

After completing this lesson, you should be able to:

- Implement process execution in an appropriate scheduling class
- Manage process scheduling priority
- Manage the scheduling class of zones
- Configure the fair share scheduler
- Monitor the fair share scheduler

Job Workflow



Agenda

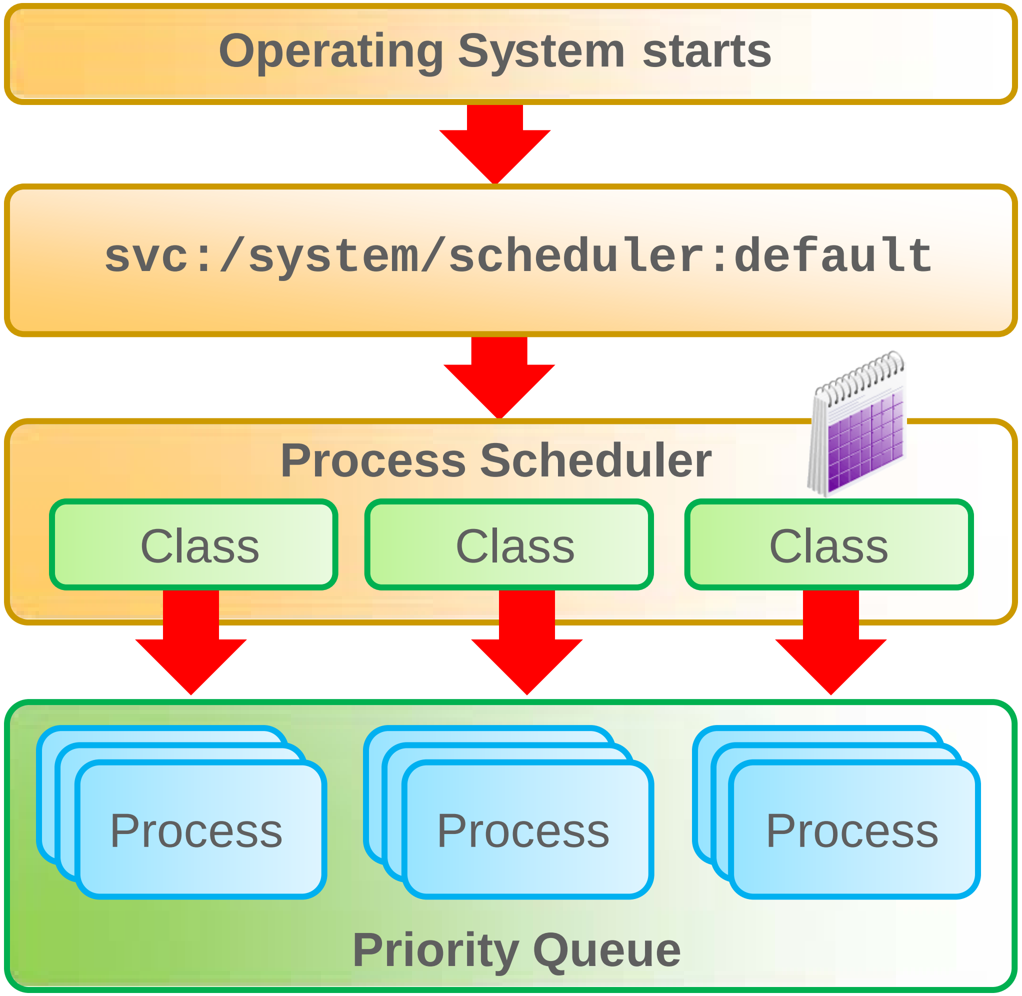
- Implementing process execution in an appropriate scheduling class
- Managing process scheduling priority
- Configuring the fair share scheduler
- Managing the scheduling class of zones

Importance of Process Execution in an Appropriate Scheduling Class

The process execution in an appropriate scheduling class plan ensures that:

- System resources are used appropriately
- Processes are prioritized in accordance with business needs and requirements
- Process workload distribution is controlled
- Processes are assigned to the appropriate scheduling class

Process Scheduler



Process Priority

- Global priority:
  - It is based on scheduling class.
- Designated priority:
  - Designating a priority affects global priority assignment and position in a priority queue.
  - Both scheduling class and user priority can be designated.
  - User priority is based on the assigned priority range of the scheduling class.

Process Scheduling Classes

Scheduling Class	Description
Time sharing (TS)	Default class for processes and their associated kernel threads. Priorities in this class are dynamically adjusted in an attempt to allocate processor resources evenly.
Interactive (IA)	Enhanced version of the TS class that applies to the in-focus window in the GUI. Its intent is to give extra resources to processes associated with that specific window.
Fair share scheduler (FSS)	This class is share based rather than priority based. Threads managed by FSS are scheduled based on their associated shares and the processor's utilization.
Fixed priority (FX)	Priorities for threads associated with this class are fixed. That is, they do not vary dynamically over the lifetime of the thread.
System (SYS)	Used to schedule kernel threads. Threads in this class are “bound” threads, which means that they run until they block or complete.
System Duty-Cycle Class (SDC)	This class is a variant of the SYS class. SDC adjusts its threads’ priority to give them a specified percentage of the processor. It also gives them a scheduling quantum, which gives other threads at the same priority on the same CPU a chance to run.
Real-Time (RT)	Threads in the RT class are fixed priority, with a fixed-time duration called quantum.

Priority Ranges for Scheduling Classes

Scheduling Class	Priority Range
Real-time (RT)	100 through 159
System (SYS)	60 through 99
Fair share scheduler (FSS)	0 through 59
Fixed priority (FX)	0 through 59
Interactive (IA)	0 through 59
Time sharing (TS)	0 through 59

Combining FSS with Other Scheduling Classes

- Ensure that the FSS, TS, IA, and FX scheduling classes do not share the same processor set (pset).
- All processes that run on a processor set must be in the same scheduling class so that they do not compete for the same CPUs.
- To avoid starving applications in the FSS class, use processor sets for FSS class and FX class applications.
- The following classes can be in the same processor sets:
  - TS and IA classes
  - FSS and RT classes

    Note: FSS has no control over the RT class processes.

Using CPU Shares with the FSS

- The FSS uses CPU shares to control the allocation of available CPU resources among workloads.
- Assigning a greater number of CPU shares to a project gives that project more CPU resources from the FSS.
- CPU share allocation and CPU resource usage are not the same.
  - CPU shares define the relative importance of workloads in relation to other workloads.
  - Resource utilization is the percentage of CPU capacity being used.
- When allocating CPU shares, you should know:
  - How many shares the project has in comparison with other projects
  - How many of the other projects are competing for CPU resources

Scheduling Class on a System with Zones Installed

- Nonglobal zones use the system’s default scheduling class.
- For a new default scheduling class setting, nonglobal zones obtain the new setting when booted or rebooted.
- To ensure that all zones get a fair share of the system CPU resources, set the FSS as the system default scheduling class.

Implementing the Process Execution in an Appropriate Scheduling Class

You will next learn how to:

- Determine the scheduling priorities and classes for the process running on the system
- Modify scheduling priorities
- Set the FSS as the default scheduler
- Configure CPU shares for zones

Quiz

For the operating system to prioritize processes, all processes must have the same scheduling class.

a. True

b. False

Quiz

Which scheduling class has the highest range of user priority designations?

a. Fair share scheduler (FSS)
b. Real-time (RT)
c. System (SYS)
d. Time sharing (TS)

Quiz

What is the purpose of CPU shares?

a. To control the allocation of available CPU resources among workloads

b. To increase CPU capacity

c. To change the global priority of a project in the priority queue

d. To cap the CPU resource usage of a process

Quiz

Nonglobal zones use the default system scheduling class for the system.

a. True

b. False

Agenda

- Implementing process execution in an appropriate scheduling class
- Managing process scheduling priority
- Configuring the fair share scheduler
- Managing the scheduling class of zones

Managing Process Scheduling Priority

This section covers the following topics:

- Displaying processes by using the `top` command
- Displaying process class information
- Determining the global priority of a process
- Designating a process priority
- Modifying a process priority

Displaying Processes by Using the `top` Command

To display the top 10 processes that are using the most CPU resources, use `top 10 –s 10`.

```
# top 10 -s 10
last pid:  1121;  load avg:  0.20,  0.14,  0.12;  up 0+01:50:30        14:10:30
87 processes: 83 sleeping, 3 running, 1 on cpu
CPU states: 81.8% idle,  5.1% user, 13.1% kernel,  0.0% iowait,  0.0% swap
Kernel: 609 ctxsw, 9 trap, 327 intr, 1935 syscall, 4 flt
Memory: 1024M phys mem, 84M free mem, 977M total swap, 977M free swap

   PID USERNAME NLWP PRI NICE  SIZE   RES STATE    TIME    CPU COMMAND
   991 oracle      2  59    0   87M   19M sleep    0:11  4.03% gnome-terminal
   733 oracle      3  59    0   65M   53M run      0:23  3.82% Xorg
   929 oracle     20  59    0  160M  140M run      2:01  1.75% java
   934 oracle      1  56    0   12M 5552K run      0:06  1.46% xscreensaver
  1120 root        1  59    0 4296K 2480K cpu      0:00  0.25% top
   917 oracle      1  49    0  107M   36M sleep    0:01  0.22% nautilus
   913 oracle      1  59    0   27M   15M sleep    0:01  0.08% metacity
    11 root       18  59    0   12M   11M sleep    0:41  0.06% svc.configd
```

Displaying Process Class Information

To display information about process classes, use `priocntl -l`.

```
# priocntl -l
CONFIGURED CLASSES
==================
SYS (System Class)

TS (Time Sharing)
Configured TS User Priority Range: -60 through 60

SDC (System Duty-Cycle Class)

FSS (Fair Share)
     Configured FSS User Priority Range: -60 through 60

FX (Fixed priority)
Configured FX User Priority Range: 0 through 60
IA (Interactive)
Configured IA User Priority Range: -60 through 60
```

Determining the Global Priority of a Process

To determine the global priority of a process, use `ps -ecl`.

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY       TIME   CMD
19 T 0   0    0    SYS 96   f00d05a8   0           ?         0:03  sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?        36:51   init
19 S 0   2    0    SYS 98   ff0f4018   0  f00c645c ?         0:01 pageout
19 S 0   3    0    SYS 60   ff0f5998   0  f00d0c68 ?       241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?         0:07    sac
 8 S 0   204  1    TS  43   ff2f6008  50  ff2f606e console   0:02     sh
```

Designating a Process Priority

1. Start a process with a designated priority by using `priocntl -e -c` *`class`* `-m` *`user-limit`* `-p` *`user-priority`* *`command-name`*.
2. Verify the process status by using `ps -ecl | grep` *`command-name`*.

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
 0 S       0  2959  2771   TS  60    ?   1865        ? pts/1       0:01 gfind
ps -ecl | grep find
 0 S       0  2959  2771   TS  60    ?   1961        ? pts/1       0:01 gfind
ps -ecl | grep find
 0 R       0  2959  2771   TS  59    ?   1985          pts/1       0:02 gfind
```

Modifying a Process Priority

1. Change the priority of the process by using `priocntl -s -p` *`user-priority pid`*.
2. Verify the process status by using `ps -ecl | grep` *`command-name`*.

```
# priocntl -s -p 30 3093
# ps -ecf | grep myprog
    root   3093   2909   RT 130 09:09:34 pts/3   0:00 /bin/bash /root/myprog
    root   3124   2771   IA  32 09:15:25 pts/1   0:00 grep myprog
```

Agenda

- Implementing process execution in an appropriate scheduling class
- Managing process scheduling priority
- Configuring the fair share scheduler
- Managing the scheduling class of zones

Configuring the Fair Share Scheduler (FSS)

This section covers the following topics:

- Making the FSS the default scheduling class
- Manually moving processes from other classes into the FSS class
- Manually moving a project's processes into the FSS class
- Tuning scheduler parameters

Making FSS the Default Scheduling Class

To set the default scheduler for the system to be FSS, use `dispadmin -d FSS`.

```
# dispadmin -d FSS
# dispadmin -d
FSS    (Fair Share)
```

```
# dispadmin -l
CONFIGURED CLASSES
==================

SYS    (System Class)
TS     (Time Sharing)
SDC    (System Duty-Cycle Class)
FSS    (Fair share)
FX     (Fixed Priority)
IA     (Interactive)
FSS    (Fair Share)
```

Manually Moving the `init` Process into the FSS Class

To move the `init` process into the FSS class, use `priocntl -s -c FSS -i pid 1`.

```
# ps -ecf | grep init
    root     1     0   TS  59 07:42:52 ?        0:00 /usr/sbin/init
    root  4564  1262  FSS   1 10:01:48 pts/1    0:00 grep init
# priocntl -s -c FSS -i pid 1
# ps -ef -o class,zone,fname | grep init
 FSS   global init
```

Manually Moving Processes from Other Classes into the FSS Class

To move all processes into the FSS class, use `priocntl -s -c FSS -i all`.

```
# priocntl -s -c FSS -i all
```

```
# ps -ef -o class,zone,fname | grep -v CLS | sort -k2 | more
 FSS   global asr-noti
 FSS   global auditd
 FSS   global automoun
 FSS   global automoun
 FSS   global bash
 FSS   global bash
 FSS   global bonobo-a
 FSS   global clock-ap
 FSS   global console-
 FSS   global cron
 FSS   global dbus-dae
```

Manually Moving a Project’s Processes into the FSS Class

To move the processes that run in a project to the FSS scheduling class, use `priocntl -s -c FSS -i projid projectID_number`.

```
# ps -o user,pid,uid,projid,project,class
    USER   PID   UID PROJID  PROJECT  CLS
    root  2771     0      1 user.root   TS
    root  3000     0      1 user.root   TS
# priocntl -s -c FSS -i projid 1
# ps -o user,pid,uid,projid,project,class
    USER   PID   UID PROJID  PROJECT  CLS
    root  2771     0      1 user.root  FSS
    root  3015     0      1 user.root  FSS
```

Changing Scheduler Parameters

To change the scheduler parameters, use `dispadmin -c scheduler -g [-r resolution]`.

```
$ dispadmin -c FSS -g
#
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

Practice 9-1 Overview: Modifying Process Scheduling Priority

This practice covers the following topics:

- Managing scheduling class and process priorities
- Configuring the fair share scheduler

Agenda

- Implementing process execution in an appropriate scheduling class
- Managing process scheduling priority
- Configuring the fair share scheduler
- Managing the scheduling class of zones

Managing the Scheduling Class of Zones

This section covers the following topics:

- Assigning CPU shares to the global zone
- Configuring CPU shares configuration in a nonglobal zone
- Measuring CPU performance in the zones
- Removing the CPU shares configuration from a zone

Assigning CPU Shares to the Global Zone

To assign CPU shares to the global zones, use `prctl -n zone.cpu-shares -v` *`number_of_shares`* `-r -i zone global`.

```
# prctl -n zone.cpu-shares -v 60 -r -i zone global
```

Configuring CPU Shares Configuration in a Nonglobal Zone

1. Add the CPU shares to the zone by using `zonecfg -z` *`zone`*.
2. Set the number of shares for the global zone by using `set cpu-shares=`*`number`*.
3. Exit `zonecfg`.
4. Verify the configuration change by using `zonecfg -z` *`zone`* `info`.

Configuring CPU Shares in a Nonglobal Zone: Example

```
# zonecfg -z hrzone
zonecfg:hrzone> set cpu-shares=80
zonecfg:hrzone> verify
zonecfg:hrzone> commit
zonecfg:hrzone> exit
# zonecfg -z hrzone info
zonename: hrzone
zonepath: /zones/hrzone
brand: solaris
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
[cpu-shares: 80]
net:
address not specified
allowed-address not specified
physical: vnic1
defrouter not specified
rctl:
name: zone.cpu-shares
value: (priv=privileged,limit=80,action=none)
(output omitted)
```

Measuring CPU Performance in the Zones

To measure CPU performance in the zones, use `prstat -Z`.

```
# prstat -Z
…
…
…
ZONEID     NPROC   SWAP    RSS MEMORY       TIME  CPU ZONE
     0        98   348M   451M     44%   0:00:50 0.3% global
     1        27    34M    43M   4.2%   0:20:09 8.3% hrzone
     2        27    34M    43M   4.2%   0:16:15 2.4% itzone
```

Removing the CPU Shares Configuration from a Zone

1. Remove the CPU shares configuration from a zone by using `zonecfg -z` *`zone`* `clear cpu-shares`.
2. Verify the configuration change by using `zonecfg -z` *`zone`* `info`.
3. Reboot the zone to make the configuration effective.

Removing the CPU Shares Configuration from a Zone: Example

```
# zonecfg -z hrzone clear cpu-shares
# zonecfg -z hrzone info
zonename: hrzone
zonepath: /zones/hrzone
brand: solaris
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
net:
    address not specified
    allowed-address not specified
     configure-allowed-address: true
    physical: vnic1
    defrouter not specified
…
…
```

Practice 9-2 Overview: Configuring the FSS in an Oracle Solaris Zone

This practice covers the following topics:

- Configuring CPU shares
- Monitoring FSS in two zones
- Removing the CPU shares configuration

Summary

In this lesson, you should have learned how to:

- Implement process execution in an appropriate scheduling class
- Manage process scheduling priority
- Manage the scheduling class of a zone
- Configure the fair share scheduler
- Monitor the fair share scheduler