OPERATING SYSTEMS

Process Management 10

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

Course Syllabus - Unit 1

PES UNIVERSITY ONLINE

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

Topics Outline

Scheduling in Linux 2.5

Scheduling in Linux 2.6.23+

Scheduling in Windows

OPERATING SYSTEMS

Linux Scheduling through version 2.5

- Prior to kernel version 2.5, ran variation of standard UNIX scheduling algorithm
- Version 2.5 moved to constant order O(1) scheduling time
  - **Preemptive, priority based**
  - Two priority ranges: time-sharing and real-time
  - Real-time range from 0 to 99 and nice value from 100 to 140
  - Map into global priority with numerically lower values indicating higher priority
  - Higher priority gets larger q
  - Task runnable as long as time left in time slice (active)
  - If no time left (expired), not runnable until all other tasks use their slices
  - All runnable tasks tracked in per-CPU run queue data structure
    - Two priority arrays (active, expired)
    - Tasks indexed by priority
    - When no more active, arrays are exchanged
- Worked well, but poor response times for interactive processes

Linux Scheduling through version 2.6.23+ Current version Linux 5.4.0-65-generic

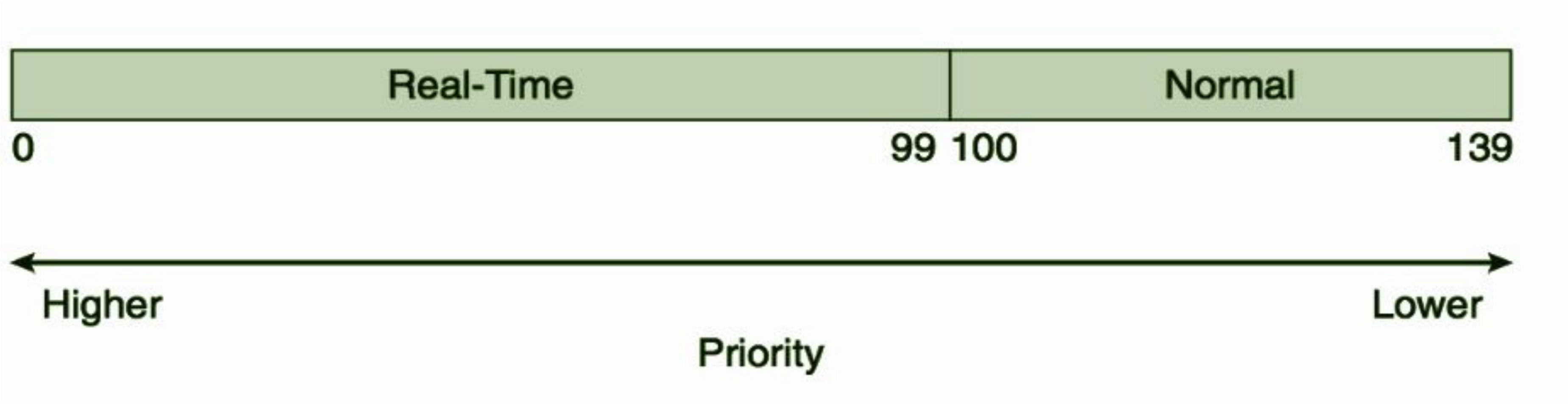
- **Completely Fair Scheduler (CFS) - Scheduling classes**
  - Each has specific priority
  - Scheduler picks highest priority task in highest scheduling class
  - Rather than quantum based on fixed time allotments, based on **proportion of CPU time**
  - Two scheduling classes included, others can be added – default and real-time
- **Quantum** calculated based on nice value from -20 to +19
  - Lower value is higher priority
  - Calculates target latency – interval of time during which task should run at least once
  - Target latency can increase if say number of active tasks increases
- CFS scheduler maintains per task virtual run time in variable vruntime
  - Associated with task priority based decay factor – lower priority is higher decay rate
  - Normal default priority yields virtual run time = actual run time
- To decide next task to run, scheduler picks task with lowest virtual run time

Finding current version of the Kernel of your host machine

- **hostnamectl**
- **cat /proc/version**
- **dmesg | grep Linux**

Linux Scheduling through version 2.6.23+

- Real-time scheduling according to POSIX.1b
- Real-time tasks have static priorities
- Real-time plus normal map into global priority scheme
- Nice value of -20 maps to global priority 100
- Nice value of +19 maps to priority 139



Linux / Unix Performance

Linux / Unix has following major resource types that need to be monitored and tuned

- CPU
- Memory
- Disk space
- Communications lines
- I/O Time
- Network Time
- Applications programs

- Try
gnome-system-monitor,
top, htop, on terminal

Linux / Unix Performance

Linux / Unix has following major resource types that need to be monitored and tuned

User State CPU	The actual amount of time the CPU spends running the users' program in the user state. It includes the time spent executing library calls, but does not include the time spent in the kernel on its behalf
System State CPU	This is the amount of time the CPU spends in the system state on behalf of this program. All **I/O routines** require kernel services. The programmer can affect this value by blocking I/O transfers
I/O Time and Network Time	This is the amount of time spent moving data and servicing I/O requests
Virtual Memory Performance	This includes context switching and swapping
Application Program	Time spent running other programs - when the system is not servicing this application because another application currently has the CPU

Linux / Unix Performance

Linux / Unix has following major resource types that need to be monitored and tuned

nice/renice	Runs a program with modified scheduling priority
netstat	Prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
time	Helps time a simple command or give resource usage
uptime	This is System Load Average
ps	Reports a snapshot of the current processes
vmstat	Reports virtual memory statistics
gprof	Displays call graph profile data
prof	Facilitates Process Profiling
top	Displays system tasks

Windows Scheduling

- Windows uses **priority-based preemptive scheduling**
- Highest-priority thread runs next
- **Dispatcher** is **scheduler**
- Thread runs until (1) blocks, (2) uses time slice, (3) preempted by higher-priority thread
- Real-time threads can preempt non-real-time
- 32-level priority scheme
- Variable class is 1-15, real-time class is 16-31
- Priority 0 is memory-management thread
- Queue for each priority
- If no runnable thread, runs idle thread

Windows Priority Class

- Win32 API identifies several priority classes to which a process can belong
- REALTIME_PRIORITY_CLASS, HIGH_PRIORITY_CLASS, ABOVE_NORMAL_PRIORITY_CLASS,NORMAL_PRIORITY_CLASS, BELOW_NORMAL_PRIORITY_CLASS, IDLE_PRIORITY_CLASS
- All are variable except REALTIME
- A thread within a given priority class has a relative priority TIME_CRITICAL, HIGHEST, ABOVE_NORMAL, NORMAL, BELOW_NORMAL, LOWEST, IDLE
- Priority class and relative priority combine to give numeric priority
- Base priority is NORMAL within the class
- If Adapted quantum expires, priority lowered, but never below base

Windows Priority Class

- If wait occurs, priority boosted depending on what was waited for
- Foreground window given 3x priority boost
- Windows 7 added **User-Mode scheduling (UMS)**
- Applications create and manage threads independent of kernel
- For large number of threads, much more efficient
- UMS schedulers come from programming language libraries like C++ Concurrent Runtime (ConcRT) framework

Windows Priorities

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

My System Screen Shots

```
sridatta@sridatta:~$ sensors
nouveau-pci-0100
Adapter: PCI adapter
GPU core:     912.00 mV (min =  +0.80 V, max =  +1.19 V)
fan1:         2730 RPM
temp1:         +45.0°C  (high = +95.0°C, hyst =  +3.0°C)
                        (crit = +105.0°C, hyst =  +5.0°C)
                        (emerg = +135.0°C, hyst =  +5.0°C)

acpitz-acpi-0
Adapter: ACPI interface
temp1:         +16.8°C  (crit = +20.8°C)
temp2:         +27.8°C  (crit = +119.0°C)
temp3:         +29.8°C  (crit = +119.0°C)

coretemp-isa-0000
Adapter: ISA adapter
Package id 0:   +37.0°C  (high = +82.0°C, crit = +100.0°C)
Core 0:         +36.0°C  (high = +82.0°C, crit = +100.0°C)
Core 1:         +37.0°C  (high = +82.0°C, crit = +100.0°C)
Core 2:         +36.0°C  (high = +82.0°C, crit = +100.0°C)
Core 3:         +34.0°C  (high = +82.0°C, crit = +100.0°C)
Core 4:         +35.0°C  (high = +82.0°C, crit = +100.0°C)
Core 5:         +35.0°C  (high = +82.0°C, crit = +100.0°C)

sridatta@sridatta:~$ 
```

OPERATING SYSTEMS

My System Screen Shots

Summary

Computer		
	Processor	Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
	Memory	32831MB (2703MB used)
	Machine Type	Desktop
	Operating System	Ubuntu 20.04.1 LTS
	User Name	sridatta (sridatta)
	Date/Time	Wednesday 26 August 2020 08:11:54 AM
Display		
	Resolution	1680x1050 pixels
	OpenGL Renderer	(Unknown)
	X11 Vendor	The X.Org Foundation
Audio Devices		
	Audio Adapter	HDA-Intel - HDA Intel PCH
	Audio Adapter	USB-Audio - USB Device 0x46d
	Audio Adapter	HDA-Intel - HDA NVidia
Input Devices		
	Sleep Button	
	Power Button	
	Power Button	
	PixArt Dell MS116 USB Optical Mouse	
	Dell KB216 Wired Keyboard	
	Dell KB216 Wired Keyboard Consumer Control	
	Dell KB216 Wired Keyboard System Control	
	UVC Camera (046d:0825)	
	HDA Intel PCH Front Mic	
	HDA Intel PCH Rear Mic	
	HDA Intel PCH Line	
	HDA Intel PCH Line Out	
	HDA Intel PCH Front Headphone	
	HDA NVidia HDMI/DP,pcm:3	
	HDA NVidia HDMI/DP,pcm:7	
Printers		
	No printers found	
SCSI Disks		
	ATA CT240BX500SSD1	
	ATA ST1000DM010-2EP1	

Operating System

Version		
	Kernel	Linux 5.4.0-42-generic (x86_64)
	Version	#46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020

My System Screen Shots

Display

Display		
	Resolution	1680x1050 pixels
	Vendor	The X.Org Foundation
	Version	1.20.8
	Current Display Name	:0
Monitors		
	Monitor 0	1680x1050 pixels
OpenGL		
	Vendor	(Unknown)
	Renderer	(Unknown)
	Version	(Unknown)
	Direct Rendering	No
Extensions		
	Composite	
	DAMAGE	
	DOUBLE-BUFFER	
	DPMS	
	DRI2	
	DRI3	
	GLX	

My System Screen Shots

```
sridatta@sridatta:~$ vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 27684300 147972 2259256    0    0    33    26  130  426  4  3 93  0  0
sridatta@sridatta:~$
```

```
sridatta@sridatta:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
CPU(s):                          6
On-line CPU(s) list:             0-5
Thread(s) per core:              1
Core(s) per socket:              6
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           158
Model name:                      Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
Stepping:                        10
CPU MHz:                         800.050
CPU max MHz:                     4100.0000
CPU min MHz:                     800.0000
BogoMIPS:                        5799.77
Virtualization:                  VT-x
L1d cache:                       192 KiB
L1i cache:                       192 KiB
L2 cache:                        1.5 MiB
L3 cache:                        9 MiB
NUMA node0 CPU(s):               0-5
```

My System Screen Shots

```
sridatta@sridatta:~$ lscpu |grep 'CPU(s)'
CPU(s):                          6
On-line CPU(s) list:             0-5
NUMA node0 CPU(s):               0-5
sridatta@sridatta:~$
```

OPERATING SYSTEMS

My System Screen Shots

Topics Uncovered in this Session

Scheduling in Linux 2.5

Scheduling in Linux 2.6.23+

Scheduling in Windows

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com