OS Lab #05 – Solution

Task 1: Viva Questions

a. What do you mean by process scheduling, and what are different types of process schedulers?

Process scheduling is the mechanism by which the operating system decides the order in which processes are executed on a CPU. It ensures that all processes get access to the CPU while balancing efficiency and fairness. The main types of process schedulers are:

- Long-term scheduler (Job Scheduler)

- Short-term scheduler (CPU Scheduler)

- Medium-term scheduler

b. Difference between short-term and long-term scheduler.

- Short-term Scheduler: Allocates CPU time to processes already in memory, managing the CPU's immediate load.

- Long-term Scheduler: Manages the selection of processes from the job pool into the ready queue, balancing load between CPU and I/O processes.

c. What are different types of scheduling algorithms that you have studied so far, and what were their flaws?

- First-Come, First-Served (FCFS): Simple but can lead to the convoy effect.

- Shortest Job Next (SJN): Minimizes average wait time but may suffer from starvation.

- Round Robin (RR): Fair for time-sharing but has higher overhead due to frequent context switching.

- Priority Scheduling: Prioritizes important tasks but can lead to starvation of lower-priority processes.

- Multilevel Queue Scheduling: Provides different types of scheduling for different types of processes but is complex to manage.

d. Describe the working of UNIX SVR3 scheduler.

The UNIX System V Release 3 (SVR3) scheduler uses a multilevel feedback queue, allowing processes to move between queues based on their behavior and CPU usage. It adjusts the priority of processes dynamically to ensure better responsiveness.

e. What is the nice value of a process?

The nice value of a process is an integer that represents the priority level of a process, ranging from -20 (most favorable) to 19 (least favorable). It helps adjust the scheduling priority of a process in a Unix-like system.

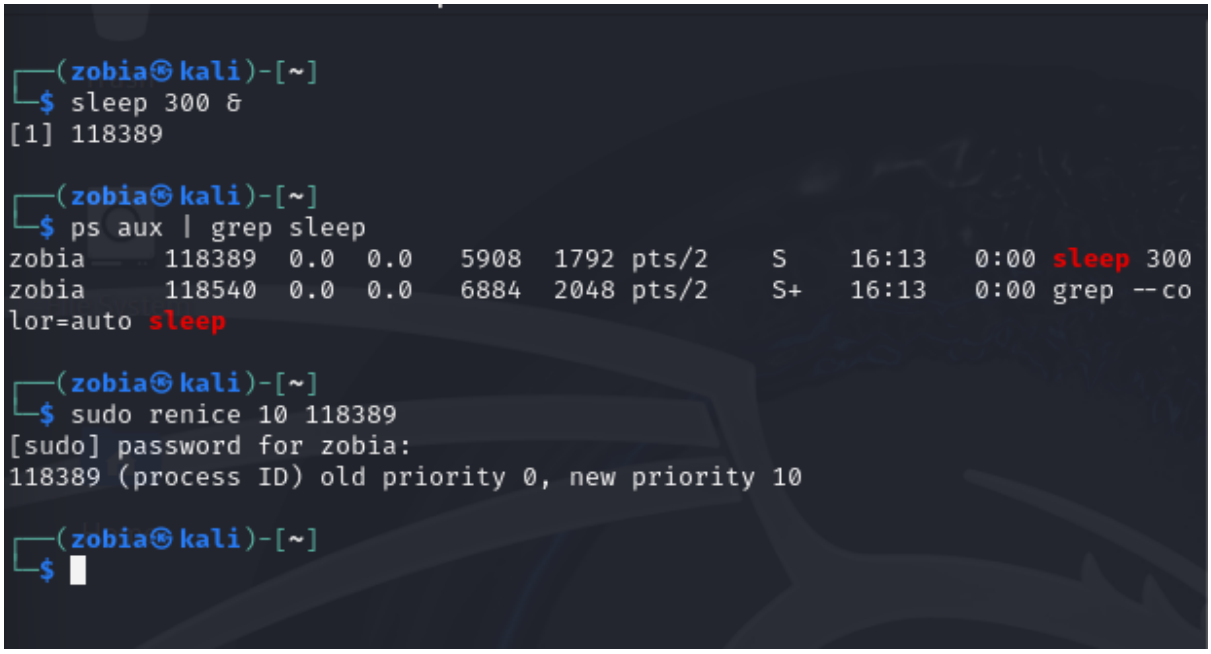
f. From where does a process get its nice value and how can we change it?

A process inherits its nice value from its parent process. It can be changed using the `nice` command at the time of execution or with the `renice` command for an already running process.

g. What is the current scheduling algorithm in Linux Kernel? What are some of its distinct features?

The current scheduling algorithm in the Linux Kernel is the Completely Fair Scheduler (CFS). It uses a red-black tree for managing processes and aims for balanced process allocation with minimal latency and fairness in CPU time distribution.

Task 2: Execute a “sleep 300” command and then change its nice value using “renice” command.



1. Started a sleep process and ran it in the background with sleep 300 &. This created a background process with PID 118389.
2. Listed the running sleep process using ps aux | grep sleep, which confirmed that the process with PID 118389 is running.
3. Changed the nice value of the sleep process from its default (0) to 10

Task 3: Execute `find` command with a nice value of -20 to find all the files whose name contain the word "libc".

Command

Using sudo is necessary because a nice value of -20 requires root privileges.

Task 4: Read the man page of `schedtool` and then tell what is it.

`schedtool` is a utility for manipulating process scheduling policies and parameters. It allows users to set different scheduling policies like SCHED_RR, SCHED_FIFO, and adjust parameters like nice value and CPU affinity for a process.

Task 5: What are different process scheduling parameters?

The different process scheduling parameters include:

- Policy (e.g., SCHED_NORMAL, SCHED_FIFO)

- Priority (static and dynamic)

- Nice value (affecting priority)

- CPU Affinity (binding a process to a specific CPU core)

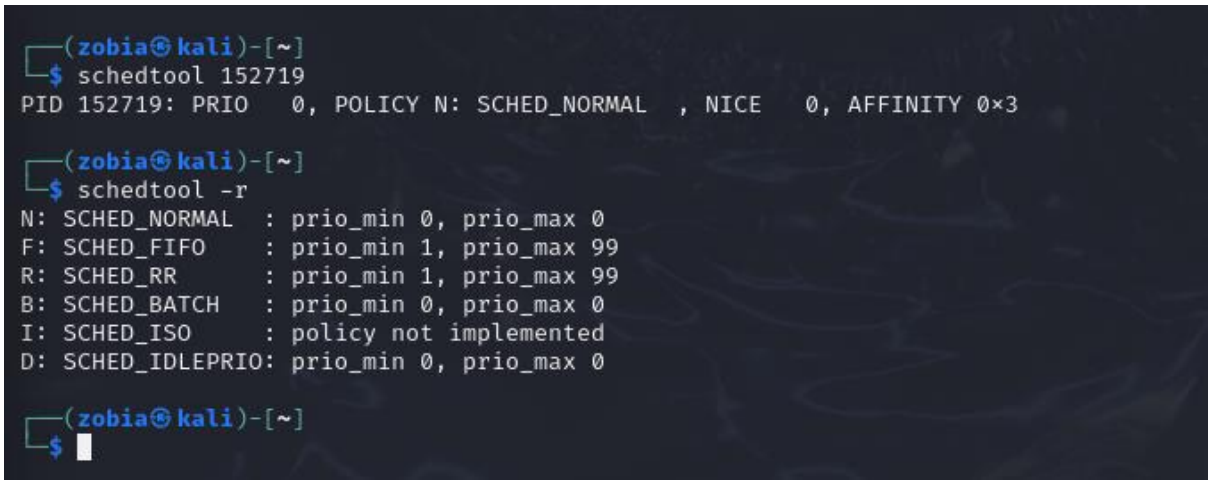
Task 6: What is CPU affinity and describe 2 of its types in Linux Kernel.

CPU affinity defines how a process is bound to a specific CPU core(s).

- Soft Affinity: The kernel attempts to keep a process on a preferred CPU but allows migration.

- Hard Affinity: A process is strictly bound to a particular CPU, and the kernel does not migrate it to another.

Task 7: Using `schedtool`, get different scheduling parameters of any process and all available scheduling policies.



Task 8: Use `schedtool` to change the following scheduling parameters of your running process and note your observations:

- Change scheduling policy to `SCHED_BATCH`, `SCHED_IDLEPRIO`, `SCHED_NORMAL`.

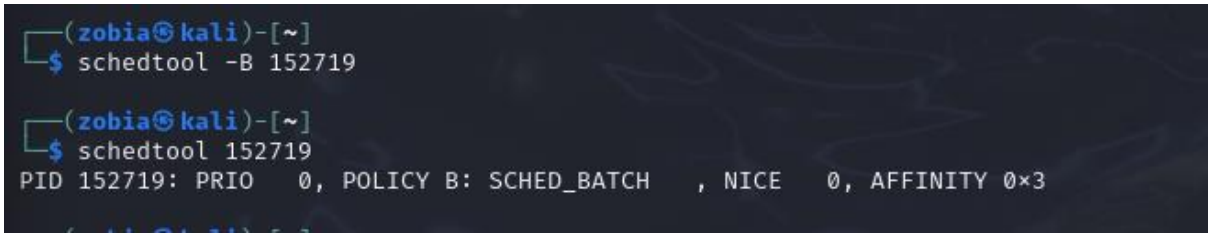
- Change nice value.

- Change static priority.

- Change CPU affinity.

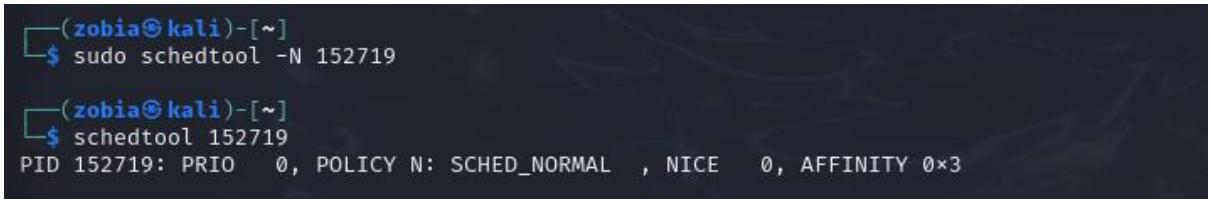
Answer:

Change scheduling policy to “SCHED_BATCH”:



Change scheduling policy to “SCHED_IDLEPRIO”:

Change scheduling policy to “SCHED_NORMAL”:



Change nice value:

Change static priority:

To change the static priority (for example, to 10 using SCHED_FIFO):

```
┌──(zobia㉿kali)-[~]
└─$ sudo schedtool -F -p 10 152719

┌──(zobia㉿kali)-[~]
└─$ schedtool 152719
PID 152719: PRIO  10, POLICY F: SCHED_FIFO    , NICE  10, AFFINITY 0×3
```

Change CPU affinity:

To set the CPU affinity (for example, to run only on CPU 0):

```
┌──(zobia㉿kali)-[~]
└─$ schedtool -a 0×1 152719

┌──(zobia㉿kali)-[~]
└─$ schedtool 152719
PID 152719: PRIO  10, POLICY F: SCHED_FIFO    , NICE  10, AFFINITY 0×1
```

Task 9: Give answers to the following questions:

a. What is proc file system and why is it called the window to the running Linux Kernel?

The “/proc” file system provides a mechanism to access kernel and process information as files. It is called the "window to the running Linux Kernel" because it allows users to interact with kernel data structures and process details.

b. Why is the size of almost all the files in this directory 0?

Files in “/proc” are virtual and dynamic, representing real-time information, rather than stored data, which is why they show a size of 0.

c. What information does the file `/proc/version` contain and which commands use this information?

“/proc/version” contains the Linux kernel version, compiler details, and build information. Commands like “uname -r” and “cat /proc/version” utilize this information.

d. What do the files `cmdline`, `environ`, `limits`, and `status` in `/proc/[PID]/` directory represent?

- “cmdline”: Command-line arguments used to start the process.

- “environ”: Environment variables for the process.

- “limits”: Resource limits set for the process.

- “status”: General information about the process, such as its state, memory usage, and thread information.

e. What can be the different types of status of a process?

- Running (R)

- Sleeping (S)

- Disk Sleep (D)

- Zombie (Z)

- Stopped (T)

- Tracing Stop (t)

f. What does the directory `/proc/[PID]/fd` contain?

It contains symbolic links to the file descriptors opened by a process, allowing access to files it has open.

g. How can you get the uptime and information about our CPU from the proc directory?

- Uptime: cat /proc/uptime

```
┌──(zobia㉿kali)-[~]
└─$ cat /proc/uptime
19885.90 34530.61
```

- CPU Information: cat /proc/cpuinfo

```
┌──(zobia㉿kali)-[~]
└─$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
model name      : 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
stepping        : 1
microcode       : 0×ffffffff
cpu MHz         : 2803.198
cache size      : 12288 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca c
mov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc
 rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse
3 cx16 pcid sse4_1 sse4_2 movbe popcnt aes rdrand hypervisor lahf_lm abm 3dn
owprefetch ibrs_enhanced fsgsbase bmi1 bmi2 invpcid rdseed clflushopt arat m
d_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs retbleed ei
brs_pbrsb bhi
bogomips        : 5606.39
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```

h. What does the directory `/proc/sys/kernel` contain?

It contains files for modifying kernel parameters at runtime, such as hostname, domain name, and process behavior.