SIC-IOT702 / task 03 : capstone Task / phase 4

Instractor : eng/ Mohamed Ahmed

Student name: Ahmed Mohamed Elsayed

Day 4 – Phase 4: Process and Network Monitoring

Boss's Request: Monitor the system while simulating sensor activity.

Tasks:

• Run a background task to simulate sensor polling.

```
zawawy34@zawawy34-VirtualBox:~$ nohup sh -c 'while true; do echo "Simulating sen
sor activity..."; sleep 10; done'
nohup: ignoring input and appending output to 'nohup.out'
```

• List processes and filter for the background task.

```
zawawy34@zawawy34-VirtualBox:~$ ps aux | grep "Simulating sensor activity"
zawawy34    2031  0.0  0.0   9228  2560 pts/0    S+   00:00   0:00 grep --color=
auto Simulating sensor activity
```

• Check network states (established connections).

```
zawawy34@zawawy34-VirtualBox:~$ ss -t -a
State      Recv-Q  Send-Q   Local Address:Port       Peer Address:Port  Process
LISTEN     0       128          127.0.0.1:ipp             0.0.0.0:*
LISTEN     0       4096     127.0.0.53%lo:domain          0.0.0.0:*
TIME-WAIT  0       0          10.0.2.15:59866    185.125.188.54:https
LISTEN     0       128            [::1]:ipp               [::]:*
```

• Try foreground and background switching.

```
zawawy34@zawawy34-VirtualBox:~$ sleep 40
^Z
[1]+  Stopped                 sleep 40
zawawy34@zawawy34-VirtualBox:~$ bg
[1]+ sleep 40 &
zawawy34@zawawy34-VirtualBox:~$ fg
sleep 40
^C
```

• Kill a process if needed.

```
zawawy34@zawawy34-VirtualBox:~$ kill 2266
[1]-  Terminated              nohup sh -c 'while true; do echo "Simulating senso
r activity..."; sleep 10; done'
zawawy34@zawawy34-VirtualBox:~$
```

Open-Ended Questions:

Bash Command Execution:

- When you type a command in Bash (e.g., ls), a series of steps occur:
    1. **Parse and Tokenize**: Bash reads the input line, breaking it down into individual words or "tokens."
    2. **Alias and History Expansion**: Bash checks if the first token is an alias and performs any history expansions (e.g., !!).
    3. **Command Lookup**: It looks for the command in the following order:
        - **Built-in Command**: If it's a built-in command (like cd or echo), Bash executes it directly.
        - **Path Search**: If not a built-in, Bash searches for the executable file in the directories specified by the $PATH environment variable.
        - **External Command**: Once the executable is found, Bash creates a new process (a child process) to run it using the fork() system call.
    4. **Execution**: The child process uses the exec() system call to replace itself with the new program (ls).
    5. **Output**: The program's output is sent to standard output (stdout), which is typically your terminal, and you see the results.

 Types of Processes:

- **Daemon Process**: A background process that runs continuously to provide a service. They are typically started at boot time and are not associated with a controlling terminal. Examples include web servers (httpd) and SSH daemons (sshd). They often have a name ending in d. You can detect them using ps aux | grep <daemon_name>.
- **Zombie Process**: A process that has finished execution but whose entry is still in the process table. This happens because the parent process hasn't yet called wait() to retrieve the child's exit status. Zombie processes don't consume system resources (other than a process table entry) and are eventually reaped by their parent or the init process. You can detect them using ps aux | grep 'Z'.
- **Orphan Process**: A process whose parent has died. The orphan process is immediately adopted by the init process (PID 1) which then takes responsibility for cleaning it up once it finishes execution. Orphan processes continue to run normally.

 Inter-Process Communication (IPC):

- We need IPC to allow different processes to exchange data and synchronize their actions. Without IPC, processes would be completely isolated and unable to collaborate on tasks, which is essential for modern operating systems and applications.
- **IPC Mechanisms**:
    - **Pipes/Named Pipes**: A simple, one-way communication channel. A pipe connects the standard output of one process to the standard input of another. Example: ls | grep .txt.
    - **Message Queues**: A list of messages that processes can send and receive.
    - **Semaphores**: Used for synchronization to control access to shared resources.
    - **Shared Memory**: The fastest IPC method, where two or more processes can access the same memory region.
    - **Sockets**: Used for communication between processes on the same machine or over a network. Example: A client-server application.