Experiment 9: Shell Programming Continued – System Performance Monitoring

---

Experiment 9: Shell Programming Continued – System Performance Monitoring

---

Name: Rishika Purushotham Roll No.: 590029145 Date: 2025-09-26

Aim:

- To continue shell programming practice with scripts for system performance monitoring.
- To use Linux commands (`top`, `htop`, `uptime`, `free`, `vmstat`, `iostat`, `sar`) to monitor CPU, memory, and disk usage.
- To automate performance checks with shell scripts.

Requirements

- A Linux machine with bash shell.
- Tools: `top`, `htop`, `vmstat`, `iostat`, `sar` (install `sysstat` package for `sar`).

Theory

System performance monitoring ensures efficient use of CPU, memory, and I/O resources. Linux provides commands like:

- `top`/`htop`: real-time CPU, memory, process usage.
- `uptime`: system uptime and load average.
- `free`: RAM and swap usage.
- `vmstat`: process, memory, CPU statistics.
- `iostat`: CPU and disk I/O statistics.
- `sar`: historical system statistics.

Shell scripts can capture these outputs into log files for analysis and troubleshooting.

Procedure & Observations

Exercise 1: Checking CPU and Memory with `top` and `htop`

Task Statement:

Run `top` and `htop` to observe CPU and memory usage in real-time.

Command(s):

```
top
htop
```

Output:

Exercise 2: System Load and Uptime

Task Statement:

Check system uptime and load averages.

Command(s):

```
uptime
```

Output:

Exercise 3: Memory Usage with `free`

Task Statement:

Display total, used, and free memory along with swap usage.

Command(s):

```
free -h
```

Output:

Exercise 4: Monitoring with `vmstat`

Task Statement:

Check system processes, memory, CPU, and I/O statistics.

Command(s):

```
vmstat 5 5
```

Output:

Exercise 5: Disk I/O Monitoring with `iostat`

Task Statement:

Monitor CPU utilization and I/O statistics of devices.

Command(s):

```
iostat -xz 5 3
```

Output:

Exercise 6: Historical Monitoring with `sar`

Task Statement:

Display CPU usage statistics using `sar`.

Command(s):

```
sar -u 5 5
```

Output:

Exercise 7: Automating Performance Logs

Task Statement:

Write a script `monitor.sh` that logs CPU, memory, and disk usage every minute.

Command(s):

```bash
#!/bin/bash
while true; do
  echo "--- $(date) ---" >> perf_log.txt
  uptime >> perf_log.txt
  free -h >> perf_log.txt
  iostat -xz 1 1 >> perf_log.txt
  echo >> perf_log.txt
  sleep 60
done
```

Output:

Result

- Practiced real-time and historical system performance monitoring.
- Automated performance logging with a shell script.

Challenges Faced & Learning Outcomes

- Challenge 1: `sar` not available by default. Solved by installing `sysstat`.
- Challenge 2: Managing continuous logging. Solved by running script in background and monitoring file size.

Learning:

- Learned to interpret CPU, memory, and disk usage with various tools.
- Automated monitoring improves system administration efficiency.

Conclusion

This experiment extended shell programming into **system monitoring** using commands and scripts, building skills for real-world system administration.