Placement Empowerment Program

Cloud Computing and DevOps Centre

Day 09 – System Inventory Report Generator

Generate a report summarizing system info such as disk usage, memory, CPU, and running processes.

Name: Hema S

Department: ECE

Introduction

In modern Linux environments, keeping track of system performance and resource usage is essential for maintaining a healthy system. System administrators and developers often need quick access to critical system metrics such as disk usage, memory consumption, CPU information, and active processes.

This Proof of Concept (PoC) focuses on building a System Inventory Report Generator using a shell script. The script automates the collection of key system information and organizes it into a readable report. This helps users perform system audits, monitor performance, troubleshoot issues, or simply maintain logs for future reference.
By the end of this PoC, you will have a working automation script that can be scheduled to run periodically using cron and log the system state without manual intervention — a fundamental DevOps and Linux administration practice.

Overview

The System Inventory Report Generator is a shell script-based automation tool designed to collect and summarize essential system information on a Linux machine.

This includes:

1. Disk usage details (available and used space)
2. Memory usage statistics (RAM and swap)
3. CPU information (architecture, cores, model)
4. Top running processes by memory usage

This PoC demonstrates how to:

1. Automate system data collection using Linux commands
2. Format the output into a human-readable report
3. Schedule the script to run at defined intervals using cron
4. Store and log system reports for future reference or audits

This tool is lightweight, customizable, and ideal for system admins, DevOps engineers, and Linux users who want to maintain visibility into their system’s health and performance.

Key steps in this PoC:

✅Open the Terminal
Launch the terminal to begin writing and executing the shell script.

✅ Create the Shell Script File
Use a text editor (e.g., nano) to create a script named system_report.sh.

✅ Write Script Logic to Collect System Info
Use Linux commands like df, free, lscpu, and ps to gather:
Disk usage
Memory usage
CPU details
Top processes by memory usage

✅Format and Save the Report
Output the collected information to a timestamped .txt file.

✅ Make the Script Executable
Use chmod +x to give the script permission to run.

✅ Execute the Script
Run the script to generate the system report manually.

Objectives :

✅**Automate System Information Collection**
Create a shell script to automatically gather key system metrics like disk, memory, CPU, and processes.

✅**Generate Readable Inventory Reports**
Output the collected information into a well-structured and timestamped text report.

✅**Improve System Visibility**
Enable users or administrators to regularly monitor the system’s health and performance.

✅**Support Scheduled Monitoring**
Integrate the script with cron to run at regular intervals (e.g., daily), ensuring ongoing monitoring without manual effort.

✅**Enhance Troubleshooting and Audit Readiness**
Maintain historical system logs that help with identifying issues, capacity planning, or system audits.

Importance :

✅ **Real-Time System Monitoring**
Regular system reports help you monitor your machine's health and identify performance issues early.

✅**Simplifies Troubleshooting**
When something goes wrong, system logs and inventory reports provide valuable data for root cause analysis.

✅ **Improves Efficiency**
Automating system reporting saves time compared to manually checking each resource using separate commands.

✅**Supports Preventive Maintenance**
Helps in detecting problems such as low disk space or high memory usage before they impact system performance.

✅**Useful in Audits and Documentation**
Maintains historical logs of system state that can be used for audits, compliance, or system change tracking.

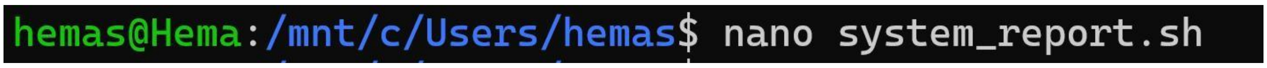
✅ **Essential for DevOps & System Admins**
Automating routine health checks is a key practice in DevOps and systems administration for scalable environments.

Step-by-Step Overview

Step 1:Open Terminal

Launch a terminal window on your Linux system.

Step 2: Create the Shell Script File

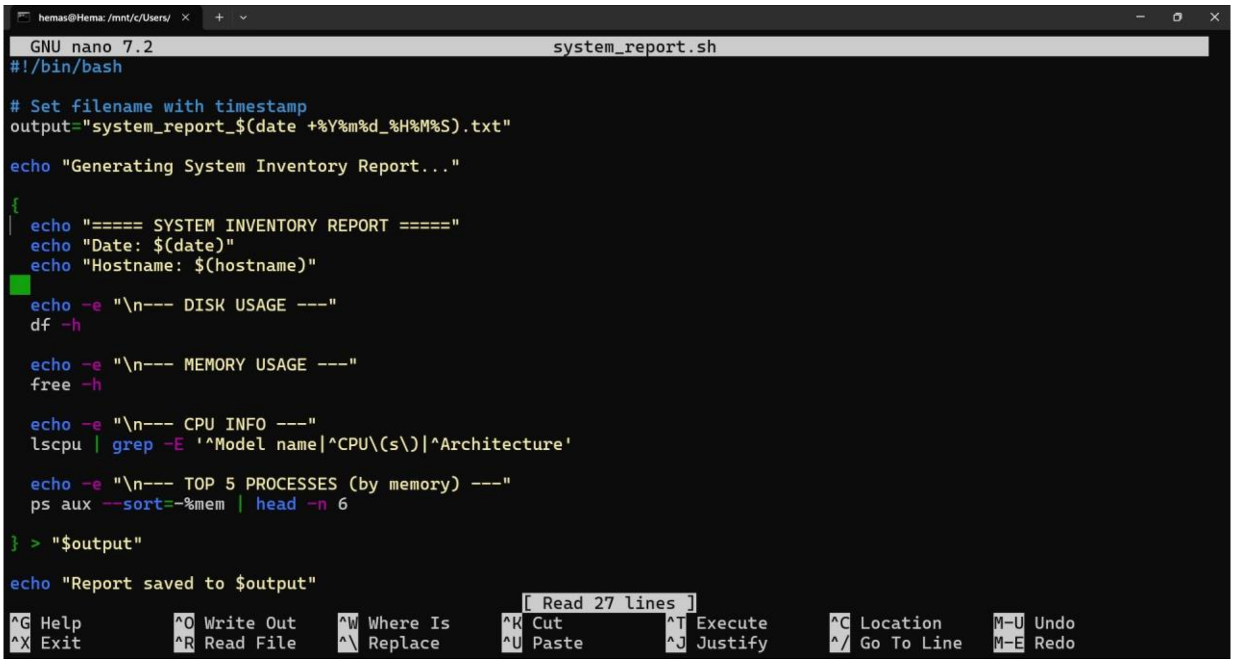


```
hemas@Hema:/mnt/c/Users/hemas$ nano system_report.sh
```

This opens the nano editor for a new file named system_report.sh.

Step 3:Write the Monitoring Script

In the nano editor,Paste the following code:



```
  GNU nano 7.2                                  system_report.sh
#!/bin/bash

# Set filename with timestamp
output="system_report_$(date +%Y%m%d_%H%M%S).txt"

echo "Generating System Inventory Report..."

{
  echo "===== SYSTEM INVENTORY REPORT ====="
  echo "Date: $(date)"
  echo "Hostname: $(hostname)"

  echo -e "\n--- DISK USAGE ---"
  df -h

  echo -e "\n--- MEMORY USAGE ---"
  free -h

  echo -e "\n--- CPU INFO ---"
  lscpu | grep -E '^Model name|^CPU\(s\)|^Architecture'

  echo -e "\n--- TOP 5 PROCESSES (by memory) ---"
  ps aux --sort=-%mem | head -n 6

} > "$output"

echo "Report saved to $output"
                                  [ Read 27 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```

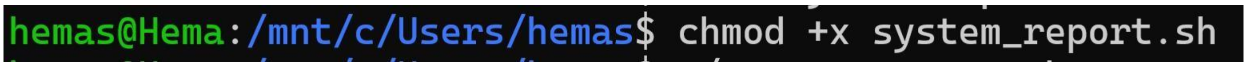
Step 4:Save and Exit

Press Ctrl + O → Enter (to save)

Press Ctrl + X (to exit)

Step 5: Make the Script Executable

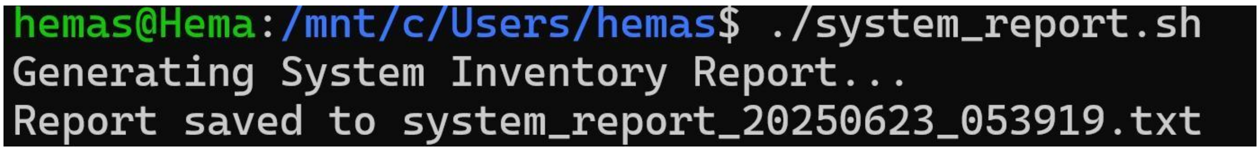
Back in the terminal:



```
hemas@Hema:/mnt/c/Users/hemas$ chmod +x system_report.sh
```

This gives the script permission to run as a program.

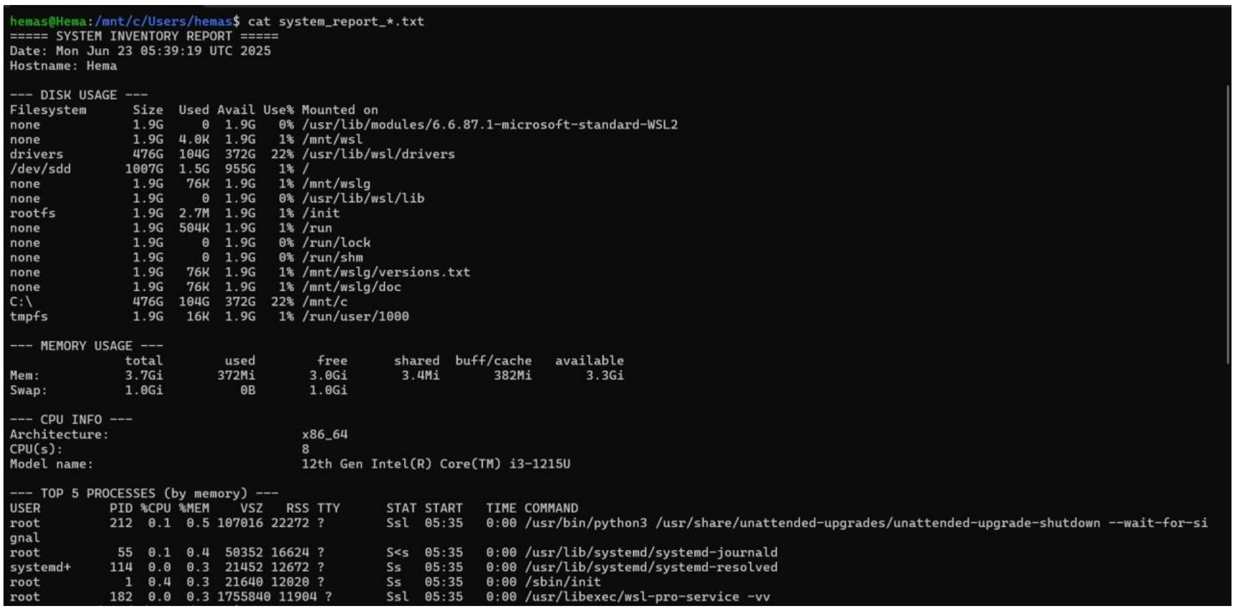
Step 6: Run the Script



```
hemas@Hema:/mnt/c/Users/hemas$ ./system_report.sh
Generating System Inventory Report...
Report saved to system_report_20250623_053919.txt
```

A new report file like system_report_20250623_053919.txt will be created.

Step 7: View the Report



```
hemas@Hema:/mnt/c/Users/hemas$ cat system_report_*.txt
===== SYSTEM INVENTORY REPORT =====
Date: Mon Jun 23 05:39:19 UTC 2025
Hostname: Hema

--- DISK USAGE ---
Filesystem      Size  Used Avail Use% Mounted on
none            1.9G     0  1.9G   0% /usr/lib/modules/6.6.87.1-microsoft-standard-WSL2
none            1.9G  4.0K  1.9G   1% /mnt/wsl
drivers         476G  104G  372G  22% /usr/lib/wsl/drivers
/dev/sdd       1007G  1.5G  955G   1% /
none            1.9G   76K  1.9G   1% /mnt/wslg
none            1.9G     0  1.9G   0% /usr/lib/wsl/lib
rootfs          1.9G  2.7M  1.9G   1% /init
none            1.9G  504K  1.9G   1% /run
none            1.9G     0  1.9G   0% /run/lock
none            1.9G     0  1.9G   0% /run/shm
none            1.9G   76K  1.9G   1% /mnt/wslg/versions.txt
none            1.9G   76K  1.9G   1% /mnt/wslg/doc
C:\             476G  104G  372G  22% /mnt/c
tmpfs           1.9G   16K  1.9G   1% /run/user/1000

--- MEMORY USAGE ---
               total        used        free      shared  buff/cache   available
Mem:           3.7Gi       372Mi       3.0Gi       3.4Mi       382Mi       3.3Gi
Swap:          1.0Gi          0B       1.0Gi

--- CPU INFO ---
Architecture:                         x86_64
CPU(s):                               8
Model name:                           12th Gen Intel(R) Core(TM) i3-1215U

--- TOP 5 PROCESSES (by memory) ---
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         212  0.1  0.5 107016 22272 ?        Ssl  05:35   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-si
gnal
root          55  0.1  0.4  50352 16624 ?        S<s  05:35   0:00 /usr/lib/systemd/systemd-journald
systemd+     114  0.0  0.3  21452 12672 ?        Ss   05:35   0:00 /usr/lib/systemd/systemd-resolved
root           1  0.4  0.3  21640 12020 ?        Ss   05:35   0:00 /sbin/init
root         182  0.0  0.3 1755840 11904 ?       Ssl  05:35   0:00 /usr/libexec/wsl-pro-service -vv
```

Outcomes:

✅ **System Inventory Shell Script Created**
You successfully created a working system_report.sh script to collect disk, memory, CPU, and process details.

✅ **Generated System Report Files**
The script creates a structured and timestamped report file every time you run it manually.

✅**Learned Key Linux Monitoring Commands**
Hands-on usage of **df, free, lscpu, ps**, and output redirection using shell scripting.

✅**Script Executable and Reusable**
You made the script executable using **chmod +x,** allowing it to be reused anytime with:

```
bash
 Copy code
 ./system_report.sh
```

✅ **Report Saved for Documentation or Debugging**
Output is saved as .txt files which can be used for system documentation, audits, or performance checks.

✅**Ready for Future Automation (Optional)**
Though you haven’t used cron yet, the script is compatible with cron jobs, so automation can be added later easily.