

Day 5 – Phase 5: Scripting Automation, Redirection & FDs

Boss's Request: Automate logging with Python and check file descriptors.

Tasks:

- Set an environment variable for sensor type.

```
export SENSOR_TYPE=temperature
echo $SENSOR_TYPE
```

```
mohamed@iot ~/iot_logger> export SENSOR_TYPE=temperature
mohamed@iot ~/iot_logger> echo $SENSOR_TYPE
temperature
```

- Write `scripts/sensor_script.py` to simulate data logging (timestamps + random values).
 - `cd iot_logger`
 - `vim scripts/sensor_script.py`
 - write script

```
mohamed@iot ~/iot_logger> cat scripts/sensor_script.py
import os, time, random

# Get sensor type from environment variable
sensor = os.getenv("SENSOR_TYPE", "unknown")

while True:
    value = random.randint(15, 45)
    print(f"{time.ctime()} | {sensor}: {value}")
    time.sleep(2)
```

- Redirect script output to `logs/temperature.log` while running as a background process.

```
python3 scripts/sensor_script.py >> logs/temperature.log &
```

```
mohamed@iot ~/iot_logger> tail -n 5 logs/temperature.log
Wed Sep 3 01:46:07 2025 | temperature: 21
Wed Sep 3 01:46:09 2025 | temperature: 26
Wed Sep 3 01:46:11 2025 | temperature: 27
Wed Sep 3 01:46:13 2025 | temperature: 26
Wed Sep 3 01:46:15 2025 | temperature: 16
```

- Find the PID of the process, inspect file descriptors in `/proc/<pid>/fd`.

```
ps -f | grep sensor_script.py
ls -l /proc/<pid>/fd
```

```
mohamed@iot ~/iot_logger> ps -f | grep sensor_script.py
mohamed 5353 3992 0 Sep02 pts/1 00:00:00 python3 scripts/sensor_script.py
mohamed 5516 3992 0 00:01 pts/1 00:00:00 grep --color=auto sensor_script.py
mohamed@iot ~/iot_logger> ls -l /proc/5353/fd
total 0
lrwx----- 1 mohamed mohamed 64 Sep 3 00:02 0 -> /dev/pts/1
l-wx----- 1 mohamed mohamed 64 Sep 3 00:02 1 -> /home/mohamed/iot_logger/logs/temperature.log
lrwx----- 1 mohamed mohamed 64 Sep 3 00:02 2 -> /dev/pts/1
```

- Filter log data into another file.

```
grep "temperature" logs/temperature.log >> logs/filtered.log
```

```
mohamed@iot ~/iot_logger> tail -n 5 logs/filtered.log
Wed Sep 3 01:47:51 2025 | temperature: 39
Wed Sep 3 01:47:53 2025 | temperature: 16
Wed Sep 3 01:47:55 2025 | temperature: 19
Wed Sep 3 01:47:57 2025 | temperature: 28
Wed Sep 3 01:47:59 2025 | temperature: 43
```

- Use wildcards to copy logs to `data/`.

```
cp logs/*.log data/
```

- Clear variable when done.

```
unset SENSOR_TYPE  
echo $SENSOR_TYPE
```

```
mohamed@iot:~/iot_logger$ unset SENSOR_TYPE  
mohamed@iot:~/iot_logger$ echo $SENSOR_TYPE  
mohamed@iot:~/iot_logger$
```

Open-Ended Questions:

- What's the difference between `' '` and `" "` in shell?
 - `' '` → everything inside is literal (no variable expansion).
 - `" "` → variables & commands inside are expanded.

```
mohamed@iot ~/iot_logger> echo '$HOME'
$HOME
mohamed@iot ~/iot_logger> echo "$HOME"
/home/mohamed
```

- Explain `[-f filename]` vs `[-d dirname]`.
 - For conditions at scripts
 - `[-f filename]` → true if file exists and is a regular file.
 - `[-d dirname]` → true if directory exists.
- Explain `stdout/stderr` redirection, appending vs overwrite. How can you confirm redirection using file descriptors?
 - `>` → overwrite file.
 - `>>` → append to file.
 - `2>` → redirect errors.
 - `&>` or `2>&1` → combine `stdout + stderr`.
 - Confirm with FDs in `/proc/<pid>/fd`:
 - `0` = `stdin`, `1` = `stdout`, `2` = `stderr`.

- Show an example of a for loop in bash. Then, write a simple bash calculator that does add/subtract.

- Script at `loop.sh`

```
mohamed@iot ~/iot> chmod +x loop.sh
mohamed@iot ~/iot> ./loop.sh
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
```

- Script at `calculator.sh`

```
mohamed@iot ~/iot> chmod +x calculator.sh
mohamed@iot ~/iot> ./calculator.sh
Simple Bash Calculator
Choose operation:
1. Add
2. Subtract
1
Enter first number:
11
Enter second number:
4

11 + 4 = 15
mohamed@iot ~/iot> ./calculator.sh
Simple Bash Calculator
Choose operation:
1. Add
2. Subtract
2
Enter first number:
22
Enter second number:
2

22 - 2 = 20
```