Name: Basant Tarik Salah

Instructor: Eng. Mohamed Abo-Khalil

SIC7_Task.Phase 5

Automate logging with Python and check file descriptors

Tasks

1. Set an environment variable for sensor type.

```
basant@basant-VirtualBox:~$ export SENSOR_TYPE="Temperature"
basant@basant-VirtualBox:~$ echo $SENSOR_TYPE
Temperature
```

2. Write scripts/sensor_script.py to simulate data logging (timestamps + random values).

```
basant@basant-VirtualBox:~$ cat > ~/iot_logger/scripts/sensor_script.py
import time, random, os

while True:
    value = random.randint(20, 30)
    timestamp = time.strftime("%Y-%m-%d %H:%M:%S")
    sensor = os.getenv("SENSOR_TYPE", "Unknown")
    print(f"{timestamp} - {sensor}: {value}")
    time.sleep(5)
```

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

```
basant@basant-VirtualBox:~$ python3 ~/iot_logger/scripts/sensor_script.py >> ~/iot_logger/logs/tempera
ture.log &
[1] 2980
```

4. Find the PID of the process, inspect file descriptors in /proc//fd.

```
basant@basant-VirtualBox:~$ ps aux | grep sensor_script.py
basant      2980  0.1  0.0  18780  8832 pts/0    S    00:54   0:00 python3 /home/basant/iot_logger/scr
ipts/sensor_script.py
basant      2982  0.0  0.0   9228  2560 pts/0    S+   00:54   0:00 grep --color=auto sensor_script.py
basant@basant-VirtualBox:~$
```

```
basant@basant-VirtualBox:~$ ls -l /proc/2980/fd
total 0
lrwx------ 1 basant basant 64 0 00:59 4  سبت -> /dev/pts/0
l-wx------ 1 basant basant 64 1 00:59 4  سبت -> /home/basant/iot_logger/logs/temperature.log
lrwx------ 1 basant basant 64 2 00:59 4  سبت -> /dev/pts/0
basant@basant-VirtualBox:~$
```

5. Filter log data into another file.

```
basant@basant-VirtualBox:~$ grep "25" ~/iot_logger/logs/temperature.log > ~/iot_logger/logs/only25.log
basant@basant-VirtualBox:~$
basant@basant-VirtualBox:~$ cp ~/iot_logger/logs/* ~/iot_logger/data/
basant@basant-VirtualBox:~$
basant@basant-VirtualBox:~$ unset SENSOR_TYPE
```

6. Use wildcards to copy logs to data/.

```
nt@basant-VirtualBox:~$ cp ~/iot_logger/logs/* ~/iot_logger/data/
nt@basant-VirtualBox:~$
```

7. Clear variable when done

```
nt@basant-VirtualBox:~$ unset SENSOR_TYPE
nt@basant-VirtualBox:~$
```

Challenge – Pipes & FD inspection

Run a pipeline (e.g., ls -l | grep .py).

While it's running, inspect the FDs in /proc//fd.

Hint: To give yourself time, put a sleep in one command of the pipeline so the process stays alive long enough for inspection.

Open Ended Questions

1. **What's the difference between ' ' and " " in shell?**

Single quotes ' ': take everything exactly as written and variables and special characters don't expand.

Double quotes " ": let variables expand as the shell replaces them with their values.

2. **Explain [-f filename] vs [-d dirname].**

They are test conditions in bash which are used inside if statements.

-f filename: checks if a file exists to do some task.

-d dirname: checks if a directory exists to do some task.

3. **Explain stdout/stderr redirection, appending vs overwrite. How can you confirm redirection using file descriptors?**

stdout/stderr redirection

Processes at Linux have 3 standard streams:

- **stdin (0):** input from keyboard.
- **stdout (1):** standard output.
- **stderr (2):** error output like error messages.

By default all of them are printed on the terminal but the **redirection** can make them be printed into specific files instead as follows:

- Redirect stdout → command > file
- Redirect stderr → command 2> file
- Redirect both → command > file 2>&1

Append vs Overwrite

> Overwrite: replaces the file with new output, erasing old content.

>> Append: adds new output at the end of the file, keeping previous content.

Confirm redirection using file descriptors

When the output is redirected the file descriptor changes.

To confirm this redirection, the process's file descriptors can be checked using: /proc/<pid>/fd/.

If **stdout** points to a file instead of the terminal, that confirms redirection.

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

```
basant@basant-VirtualBox:~$ for i in 1 2 3 4 5
do
  echo "Number $i"
done
Number 1
Number 2
Number 3
Number 4
Number 5
```

```
basant@basant-VirtualBox:~$ echo "Enter first number:"
read a
echo "Enter second number:"
read b
echo "Enter operation (+ or -):"
read op

if [ "$op" = "+" ]
then
  echo "Result = $((a + b))"
elif [ "$op" = "-" ]
then
  echo "Result = $((a - b))"
else
  echo "Unknown operation"
fi
Enter first number:
5
Enter second number:
8
Enter operation (+ or -):
+
Result = 13
basant@basant-VirtualBox:~$
```