```
GNU nano 6.2                          sensor_script.py *
#!/usr/bin/env python3
import os
import time
import random
sensor_type = os.getenv("SENSOR_TYPE", "unknown")

while True:
    t = time.ctime()
    v = random.randint(20, 30)
    print(t, "|", sensor_type, "|", v)
    time.sleep(2)
```

```
^G Help        ^O Write Out   ^W Where Is    ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File    ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

```
noor@NOOR:~/iot_logger/scripts$ nano sensor_script.py
noor@NOOR:~/iot_logger/scripts$ ls -l
total 8
-rwxrwxr-x 1 noor noor 231 21:17 3 سبت sensor_script.py
-rwxrwxr-x 1 noor noor 156 18:47 2 سبت sensor.sh
noor@NOOR:~/iot_logger/scripts$ export SENSOR_TYPE=temperature
noor@NOOR:~/iot_logger/scripts$ ./sensor_script.py
Wed Sep  3 21:18:57 2025 | temperature | 21
Wed Sep  3 21:18:59 2025 | temperature | 23
Wed Sep  3 21:19:01 2025 | temperature | 27
^CTraceback (most recent call last):
  File "/home/noor/iot_logger/scripts/./sensor_script.py", line 11, in <module>
    time.sleep(2)
KeyboardInterrupt

noor@NOOR:~/iot_logger/scripts$ ./sensor_script.py > ~/iot_logger/logs/temperature.log &
[3] 3734
noor@NOOR:~/iot_logger/scripts$ ps aux | grep sensor_script.py
noor        3685  0.0  0.1  18780  8960 pts/0   S    21:04   0:00 python3 ./sensor_scrip
t.py
noor        3704  0.0  0.1  18784  8832 pts/0   S    21:07   0:00 python3 ./sensor_scrip
t.py
noor        3734  0.0  0.1  18780  8704 pts/0   S    21:21   0:00 python3 ./sensor_scrip
t.py
```

```
noor@NOOR:~/iot_logger/scripts$ ./sensor_script.py > ~/iot_logger/logs/temperature.log &
[3] 3734
noor@NOOR:~/iot_logger/scripts$ ps aux | grep sensor_script.py
noor        3685  0.0  0.1  18780  8960 pts/0    S    21:04   0:00 python3 ./sensor_scrip
t.py
noor        3704  0.0  0.1  18784  8832 pts/0    S    21:07   0:00 python3 ./sensor_scrip
t.py
noor        3734  0.0  0.1  18780  8704 pts/0    S    21:21   0:00 python3 ./sensor_scrip
t.py
noor        3736  0.0  0.0   9228  2560 pts/0    S+   21:21   0:00 grep --color=auto sens
or_script.py
noor@NOOR:~/iot_logger/scripts$ kill -9 3685
noor@NOOR:~/iot_logger/scripts$ kill -9 3704
[1]   Killed                  ./sensor_script.py > ~/iot_logger/logs/temperature.log
noor@NOOR:~/iot_logger/scripts$ ps aux | grep sensor_script.py
noor        3734  0.0  0.1  18780  8704 pts/0    S    21:21   0:00 python3 ./sensor_scrip
t.py
noor        3740  0.0  0.0   9228  2560 pts/0    S+   21:22   0:00 grep --color=auto sens
or_script.py
[2]-  Killed                  ./sensor_script.py >> ~/iot_logger/logs/temperature.log 2>&
1
noor@NOOR:~/iot_logger/scripts$
```

```
1
noor@NOOR:~/iot_logger/scripts$ ls -l /proc/3734/fd
total 0
lrwx------ 1 noor noor 64 0 21:25 3 سبت -> /dev/pts/0
l-wx------ 1 noor noor 64 1 21:25 3 سبت -> /home/noor/iot_logger/logs/temperature.log
lrwx------ 1 noor noor 64 2 21:25 3 سبت -> /dev/pts/0
noor@NOOR:~/iot_logger/scripts$ grep "temperature" ~/iot_logger/logs/temperature.log > ~/
iot_logger/logs/filtered.log
noor@NOOR:~/iot_logger/scripts$ cat ~/iot_logger/logs/filtered.log
Wed Sep  3 21:21:07 2025 | temperature | 23
Wed Sep  3 21:21:09 2025 | temperature | 23
Wed Sep  3 21:21:11 2025 | temperature | 22
Wed Sep  3 21:21:13 2025 | temperature | 23
Wed Sep  3 21:21:15 2025 | temperature | 30
Wed Sep  3 21:21:17 2025 | temperature | 30
Wed Sep  3 21:21:19 2025 | temperature | 29
Wed Sep  3 21:21:21 2025 | temperature | 27
Wed Sep  3 21:21:23 2025 | temperature | 20
Wed Sep  3 21:21:25 2025 | temperature | 29
Wed Sep  3 21:21:27 2025 | temperature | 24
Wed Sep  3 21:21:29 2025 | temperature | 22
Wed Sep  3 21:21:31 2025 | temperature | 21
Wed Sep  3 21:21:33 2025 | temperature | 21
Wed Sep  3 21:21:35 2025 | temperature | 26
```

```
noor@NOOR: ~/iot_logger/scripts

Wed Sep  3 21:27:12 2025 | temperature | 21
Wed Sep  3 21:27:14 2025 | temperature | 24
Wed Sep  3 21:27:16 2025 | temperature | 28
Wed Sep  3 21:27:18 2025 | temperature | 24
noor@NOOR:~/iot_logger/scripts$ cp ~/iot_logger/logs/* ~/iot_logger/data/
noor@NOOR:~/iot_logger/scripts$ ls -l ~/iot_logger/data/
total 44
-rw-rw-r-- 1 noor       noor       8184 21:35 3 سبت filtered.log
-rwxrw---- 1 developer iot_team 12813 23:17 31 أكت services
-rwxrw---- 1 noor       noor      16368 21:35 3 سبت temperature.log
-rwxrw---- 1 noor       noor        156 21:35 3 سبت temperature.log.save
lrwxrwxrwx 1 developer iot_team    33 23:23 31 أكت temperature_soft.log -> ./iot_logger/l
ogs/temperature.log
noor@NOOR:~/iot_logger/scripts$ ls -l ~/iot_logger/logs/
total 28
-rw-rw-r-- 1 noor       noor       8184 21:31 3 سبت filtered.log
-rwxrw---- 1 developer iot_team 16368 21:33 3 سبت temperature.log
-rwxrw---- 1 noor       noor        156 18:45 2 سبت temperature.log.save
noor@NOOR:~/iot_logger/scripts$ echo $SENSOR_TYPE
temperature
noor@NOOR:~/iot_logger/scripts$ unset SENSOR_TYPE
noor@NOOR:~/iot_logger/scripts$ echo $SENSOR_TYPE

noor@NOOR:~/iot_logger/scripts$
```



```
noor@NOOR:~/iot_logger/scripts$ cd ~
noor@NOOR:~$ (ls -l | grep .py; sleep 20) &
[4] 3819
noor@NOOR:~$ ls -l /proc/3819/fd
ls: cannot access '/proc/3819/fd': No such file or directory
[4]+  Done                  ( ls --color=auto -l | grep --color=auto .py; sleep 20 )
noor@NOOR:~$ ls -l | (sleep 20; grep .py) &
[4] 3827
noor@NOOR:~$ ls -l /proc/3827/fd
total 0
lr-x------ 1 noor noor 64 0 21:45 3  سبت -> 'pipe:[21416]'
lrwx------ 1 noor noor 64 1 21:45 3  سبت -> /dev/pts/0
lrwx------ 1 noor noor 64 2 21:45 3  سبت -> /dev/pts/0
lrwx------ 1 noor noor 64 255 21:45 3  سبت -> /dev/pts/0
noor@NOOR:~$
```

What's the difference between ' ' and " " in shell?

' ': sees all special characters as a regular one

 ex: echo 'hello $world'   out: hello $world

" " : sees all special characters as a regular one except for $ and \

Ex : var=world   echo "hello $var"   out : hello world

Explain [-f filename] vs [-d dirname]

[-f filename] : checks if the file exists and is a regular file

[-d dirname] : check if the path exists as a directory

Explain stdout/stderr redirection

Stdout : standard output of a command

Stderr : standard error of a command

Redirection : stdout = > overwrite for ex : ls > out.txt   this overwrite the list output in the file out.txt

Another ex : ls >> out.txt   this append the output of the list to the end of the out.txt file without having to overwrite on it

Stderr = ex: ls 2> err.txt   this is error redirection and the number 2 represent the standard error of a command

appending vs overwrite

overwrite (>) : replace the file content with new output

ex :

echo first > out.txt

echo second >out.txt

 cat out.txt   the output will be :   second

append (>) : add the new output at the end of the file without deleting anything

ex :

echo first >> log.txt

echo second >> log.txt

cat log.txt   the output will be : first   second

How can you confirm redirection using file descriptors

File descriptors: 0   stdin (input)

1 stdout (output)

2 stderr (error)

Normally the FDs point to the terminal (/dev/pts/) but with redirection they point to a file   and to check the redirection of the file we may use this line : ls -l /proc/<pid>/fd

For example:

0 -> /dev/pts/0   : this means the input still point to the terminal

1 -> /home/ noor/out.txt

2 -> /home/noor/err.txt   : the output and the error points towards files