Lab 6

Distributed Software Systems

UNIBO - 2024/25

Anouk Wuyts - 1900124167

Davide De Rosa - 1186536

Linux Daemon

Background process that runs on a **Linux** (or **Unix**-**like**) system, typically without direct interaction from users.

Daemons handle tasks such as managing system services, performing maintenance operations, or responding to network requests.

They are crucial to the system's operation and often run continuously from system startup to shutdown.

Linux Daemon

Some key characteristics of Linux daemons are:

- **Background Operation**: daemons run in the background, meaning they don’t have a terminal or GUI associated with them and are not directly controlled by the user
- **System-Wide Services**: many daemons provide essential system services such as managing hardware, handling network communications or managing scheduled tasks
- **Naming Convention**: in Linux, daemons often have names that end with the letter **d** to signify that they are daemon processes

Linux Daemon in Python

The script will be able to monitor a specific directory given by the user for new files.

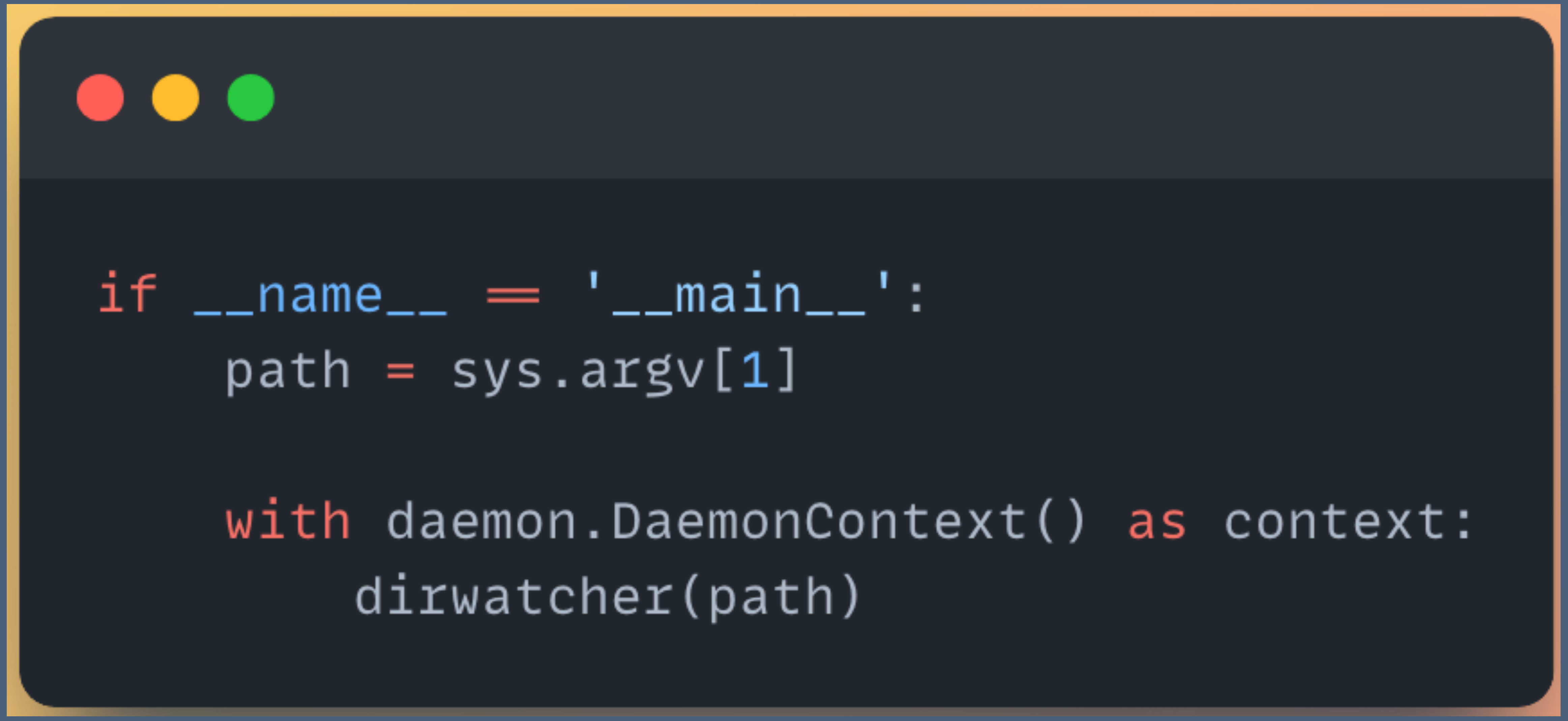
When a new file is added to the directory, a message with a timestamp is logged inside a log file.

The script is going to be implemented via the *python-daemon* library, detaching the script from the terminal and allowing us to run it as a **background daemon**.

Implementation

The directory to watch is passed as an argument by the user when executing the script.

Using the *python-daemon* library, the script is launched as a **daemon** in the background, detaching from the terminal.



```python
if __name__ == '__main__':
    path = sys.argv[1]

    with daemon.DaemonContext() as context:
        dirwatcher(path)
```

Implementation

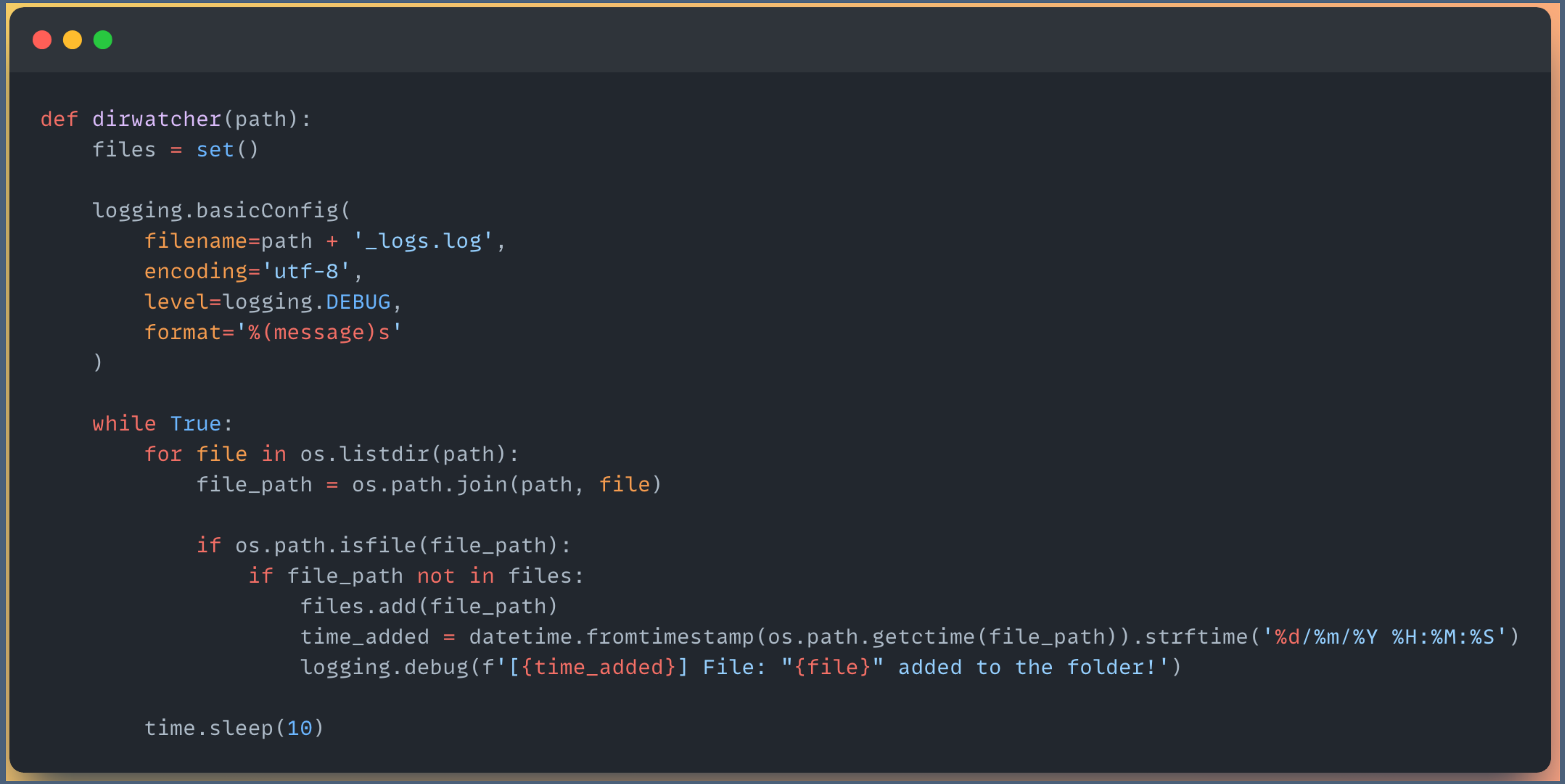
The core function of the script defines a *set* of files and setups the logging configuration.

In a *while true loop* the daemon monitors the directory given by the user and looks for new files added.

The *set* is used to check if the files monitored are new files or not.

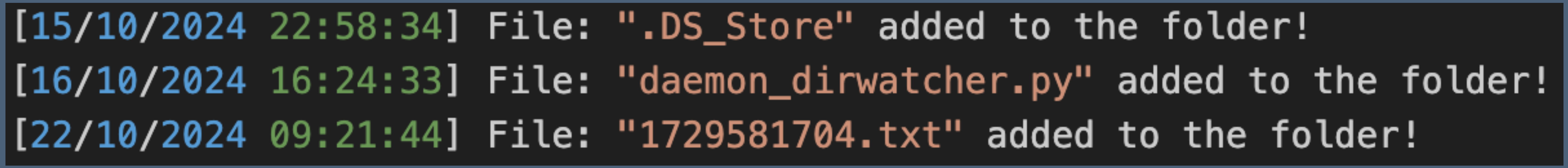
Once a new file is found, it is logged on the logs file with a timestamp.

After every execution, the daemon sleeps for 10 seconds before checking again.



```python
def dirwatcher(path):
    files = set()

    logging.basicConfig(
        filename=path + '_logs.log',
        encoding='utf-8',
        level=logging.DEBUG,
        format='%(message)s'
    )

    while True:
        for file in os.listdir(path):
            file_path = os.path.join(path, file)

            if os.path.isfile(file_path):
                if file_path not in files:
                    files.add(file_path)
                    time_added = datetime.fromtimestamp(os.path.getctime(file_path)).strftime('%d/%m/%Y %H:%M:%S')
                    logging.debug(f'[{time_added}] File: "{file}" added to the folder!')

        time.sleep(10)
```

Usage



```
[15/10/2024 22:58:34] File: ".DS_Store" added to the folder!
[16/10/2024 16:24:33] File: "daemon_dirwatcher.py" added to the folder!
[22/10/2024 09:21:44] File: "1729581704.txt" added to the folder!
```

As you can see, the script was running for different days, monitoring the directory for new files.

When a new file is being added, a line inside the logs file is printed.

Thank you!