ECS518U Operating Systems

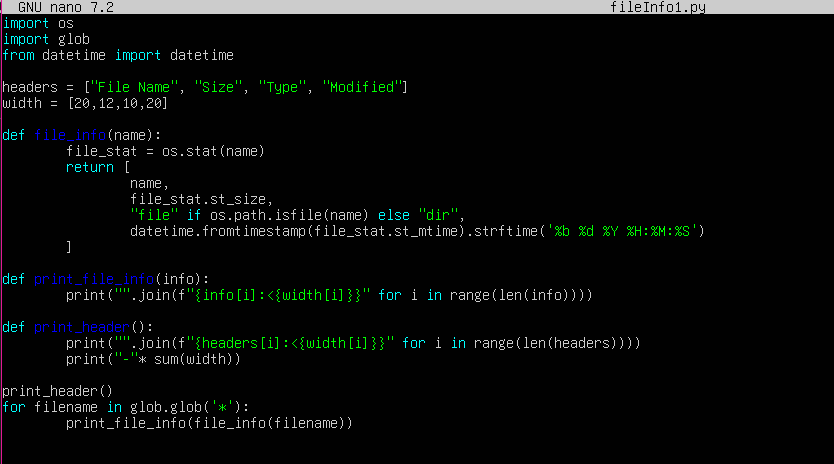
Lab 4:Filesystems (inodes & links)

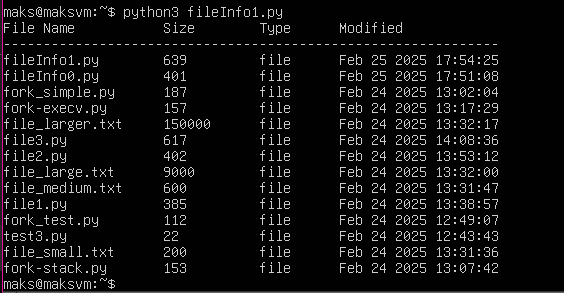
# Answer Sheet

## Step 1: Basic File Information

1. Make the changes required to fileInfo0.py to get fileInfo1.py as per the requirements. Show your code and run the script.

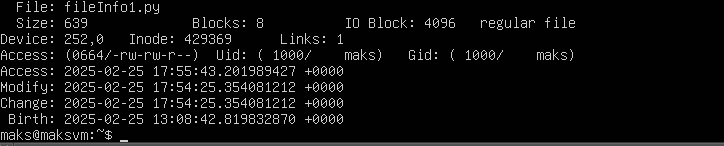
Demonstrate this by running the script and explaining your code during your lab. No need to write anything here.





1. Describe the **three different types of time information stored in the inode**. Which times are affected if you edit a file? Which times are affected if you change access permissions to a file?

There are three timestamps contained in the inode. The first being the AccessTime which checks when the file was last opened and read. The second being the Modification time which checks when changes were last made to the file. Finally the last timestamp is the changeTime which checks when the metadata of the file has been altered (includes permissions or moving the file to a new directory).



## Step 2: Links

1. Explain the difference between Hard and Symbolic Links.

|  |  |  |
| --- | --- | --- |
| **Question** | **Symbolic Link** | **Hard Link** |
| What **file type** is each link identified as in your scripts? | Link | File |
| Explain why. | A symbolic link is a separate file that stores a reference to the target file. | A hard link is an exact duplicate reference to the same file on the disk. |
| What do you observe with regards to the **inode numbers** of the target file and the new symbolic or hard link? | A new inode is created. | The same inode is retained between the hard link and the target file. |
| Explain why. | A symbolic link is just a reference to the target files location on the disk rather than a duplicate of the file. Therefore it has a different inode to the original. | A hard link points to the exact same inode as the original file meaning they are treated as the exact same file by the system. |

1. **What do you think is stored inside a symbolic link file**? What information about the symbolic link file can provide evidence about this (e.g. in the output of a shell command such as ls –l)?

The symbolic link only stores a reference to the target files location and not the actual contents of the file meaning that only the path is returned. When you use the ls –l command it only returns the symbolic link pointing to the target file.

1. **Edit the target file of a hard and of a symbolic link.** Using the stat shell command see what effect the edit has on the three time types in the inode of the hard and of the symbolic link. **Explain what you observe.**

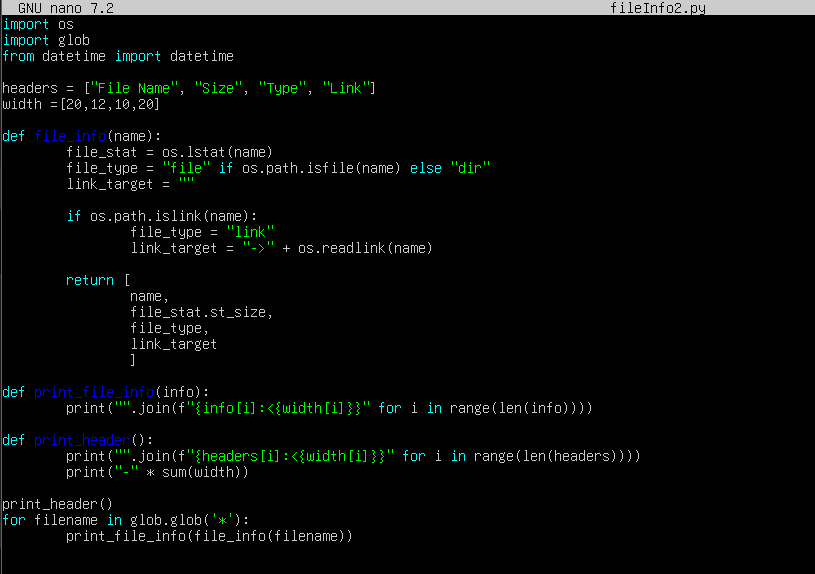
When editing a file such as LinkExampleFile.txt (seen below) , the hard link file will have its modification time and change time updated due to it sharing the same inode as the original file. However the symbolic link file will remain unchanged because it contains a different inode to the original file and only stores the reference to the original file. If LinkExampleFile.txt was deleted the hardlink would remain but the symbolic link would be broken.

## 

## Step 3: Getting information about symbolic (soft) links

Make the changes required to fileInfo1.py to get fileInfo2.py as per the requirements. Show your code and run the script.

Demonstrate this by running the script and explaining your code during your lab. No need to write anything here.



## 