

Moustafa

Family Name

3648276

Student Number

Mahmoud

Given Name

Mark:

23

[1] 1a Two file systems:

The file system type of home is nfs.
NFS stands for network file system
NFS allows users on client computers to
access files on a network,
making it a distributed file system

The file system type of root is xfs. XFS is high-
performance 64-bit journaling file system created by Silicon Graphics, Inc.
It supports metadata journaling which facilitates quicker crash recovery.

```
[mmoustaf@remotelabm39 L4]$ stat -f ~
File: "/home1/ugrads/mmoustaf"
ID: 0      Namelen: 255      Type: nfs
Block size: 32768      Fundamental block size: 32768
Blocks: Total: 134184960 Free: 41092842 Available: 41092842
Inodes: Total: 858993408 Free: 844918043
[mmoustaf@remotelabm39 L4]$ stat -f /
File: "/"
ID: fd00000000000000 Namelen: 255      Type: xfs
Block size: 4096      Fundamental block size: 4096
Blocks: Total: 26201600 Free: 12798841 Available: 12798841
Inodes: Total: 52428800 Free: 51416941
[mmoustaf@remotelabm39 L4]$
```

[1] 1b. block sizes:

The block size of the home (NFS) is 32768 bytes

The block size of the root (XFS) is 4096 bytes.

They are different because the XFS journal can be stored within the data section of the file system or on a separate device to minimize disk contention. That journal primarily contains entries that describe the portions of the disk blocks changed by file system operations. The NFS has a bigger block size due to the fact that it is networked

[2] 2a. inodes:

I noticed that the inode number of the tmp1 file is identical to that of the tmp1-hardlink file. This is not the case for the tmp1-slink file however. Its inode number is different from those of the tmp1-hardlink and tmp1 files.

This says that the hard link is a direct reference to a file using its inode. In other words, I would be able to change the content of the original file or its location and the hardlink will still point to the original file because its inode is still pointing to that file. It basically point to the exact spot on a hard drive where the inode stores the data.

On the other hand, the soft link has a separate inode value that point to the original file (like a shortcut to a file [maybe as I understand it])

```
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1
file tmp1 has inode #40799687
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1-hardlink
file tmp1-hardlink has inode #40799687
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1-slink
file tmp1-slink has inode #79345426
[mmoustaf@remotelabm39 L4]$
```

[2] 2b. hard links:

```
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1
file tmp1 has inode #40799687
file tmp1 has #2 hardlinks
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1-slink
file tmp1-slink has inode #79345426
file tmp1-slink has #1 hardlinks
[mmoustaf@remotelabm39 L4]$ ./checkFile tmp1-hardlink
file tmp1-hardlink has inode #40799687
file tmp1-hardlink has #2 hardlinks
[mmoustaf@remotelabm39 L4]$ ./checkFile checkF
checkFile* checkFile.c
[mmoustaf@remotelabm39 L4]$ ./checkFile checkFile.c
file checkFile.c has inode #281781
file checkFile.c has #1 hardlinks
[mmoustaf@remotelabm39 L4]$
```

I found that tmp1 has 2 hardlinks. tmp1-hardlink has 2 hardlinks as well. tmp1-slink has 1 hardlink. checkFile.c also has 1 hardlink.

This shows that once a file is created it automatically has a hardlink which makes sense because it will need to have a direct reference to itself. Therefore, the number of hardlinks that a file has once it is created is 1. If we create other hardlinks to that file, they will be added to that number.

[3] 3a. blocks used:

```
[mmoustaf@remotelabm39 3413mmoustaf]$ ./checkFile 512
file 512 has inode #16832657
file 512 has #1 hardlinks
file 512 uses #8 blocks
[mmoustaf@remotelabm39 3413mmoustaf]$ ./checkFile 2k
file 2k has inode #16832659
file 2k has #1 hardlinks
file 2k uses #8 blocks
[mmoustaf@remotelabm39 3413mmoustaf]$ ./checkFile 4k
file 4k has inode #16832665
file 4k has #1 hardlinks
file 4k uses #8 blocks
[mmoustaf@remotelabm39 3413mmoustaf]$ ./checkFile 4k-plus
file 4k-plus has inode #16837443
file 4k-plus has #1 hardlinks
file 4k-plus uses #16 blocks
[mmoustaf@remotelabm39 3413mmoustaf]$ ./checkFile tmp1
file tmp1 has inode #16832652
file tmp1 has #1 hardlinks
file tmp1 uses #0 blocks
```

512 has 8 blocks

2k has 8 blocks

4k has 8 blocks

4k-plus has 16 blocks

tmp1 has 0 blocks. Makes sense as the size of tmp1 is 0 bytes. Therefore, it doesn't need to use any blocks

That means any file whose size > 0 will be assigned multiples of 8 as the size increases

[1] 3b. system block size: (hand in final checkFile.c)

Due to the 8 blocks, file size to be 4096B. If and when a file is greater than that, the file as mentioned, will be assigned additional 8 blocks

[2] 3c. 24 blocks:

8192B

[1] 3d. impact of block size:

It wastes space on the HD.

[2] 3e. home vs local blocks:

[1] 4a. large file:

16 Blocks

[4] 5a. what's in a link (hand in your checkLink.c):

As seen when running the program, if we input tmp1-hardlink, we are given an error. However, that is not the case when using the tmp1-slink as the input file (Error free).
