Environment: I'm running and have been working on this project on a CentOS Linux 7.8 system via MobaXterm.

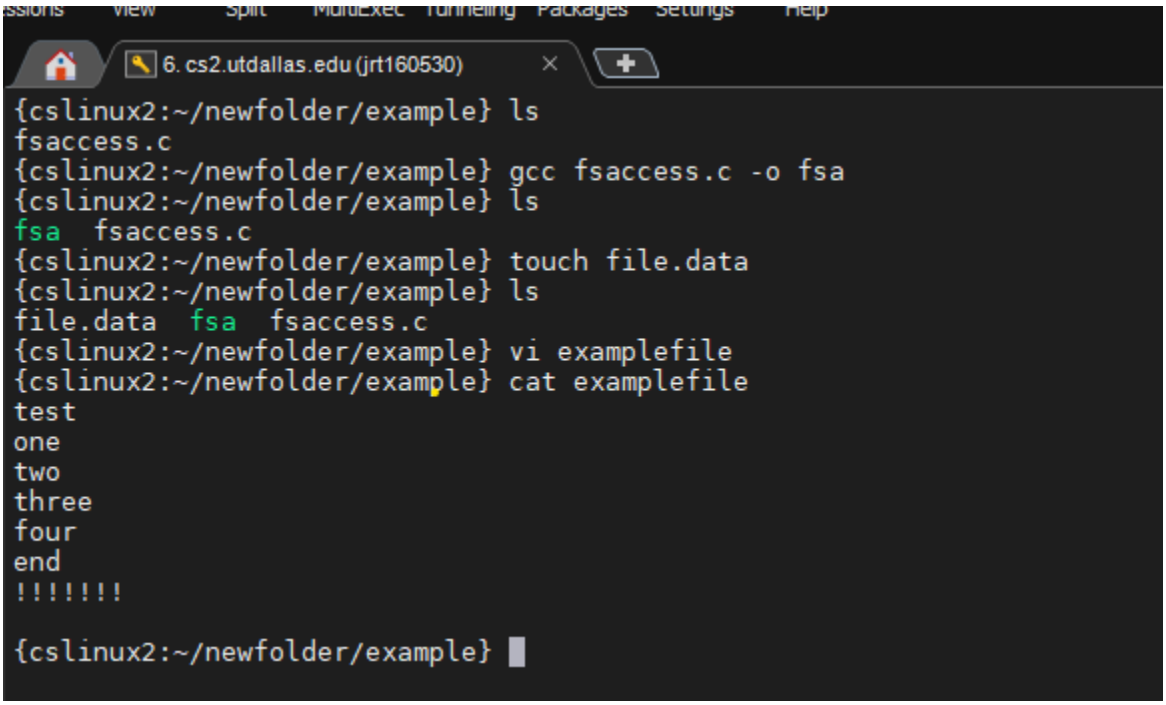
Purpose: This document will show an example of how the program is run.

Background: This project was written for an Operating System course. There will be many areas of improvement in this program. I had previous experience writing code in C, but many of the concepts, libraries, and methods were new to me and were done within a pretty narrow window of time.

Refer to the "*READ ME.txt*" document for more information.

---

1. Initially I upload and compile the program fsaccess.c and created the necessary files.
   - "file.data" is used as the mounted file system.
   - "examplefile" will be used as the file we will copy into the file system and afterwards back out.



2. Run the compiled program named "fsa". Upon running the program, the user will be prompted with a list of commands.

```
{cslinux2:~/newfolder/example} ./fsa
Type the command followed by the arguements.
For example:
mount file.data
initfs 100 10
cpin filetocopyin copiedfilenewname



(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q

$
```

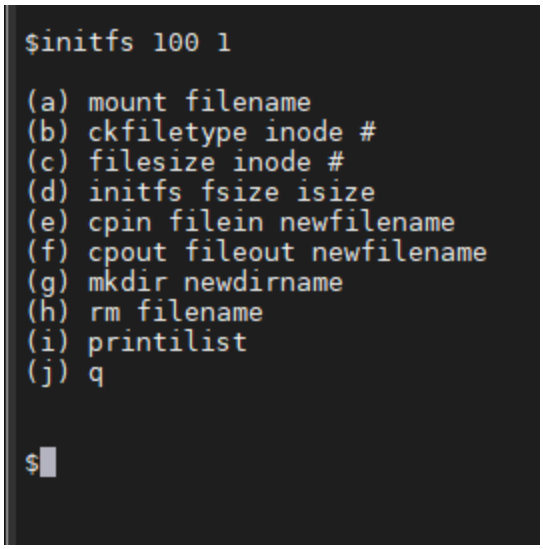
3. The first thing that needs to be done is to mount the file system. In this example, “file.data” is our file system.

```
$mount file.data
mountfunction: open successful

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$
```

4. After the file system is mounted, initialize the filesystem memory size (fsize) and size of ilist (isize).



5. After the file system is initialized, we can check the ilist with “printilist”, copy files in, copy files out, and quit and reopen the program to the saved file system (after we remount, with the same mount command above). In the proceeding pictures we will show these examples.

```
$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..

inode 2: plain file
unallocated
inode 3: plain file
unallocated
inode 4: plain file
unallocated
inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$
```

File “examplefile” copied in. I gave it the name “filein”.

I then used “pintilist” to show the update ilist.

```
$cpin examplefile filein

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 2      File name: filein

inode 2: plain file
is allocated

inode 3: plain file
unallocated
inode 4: plain file
unallocated
inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

“mkdir dir1” to make a directory name dir1 and then “printilist” to show results. We will continue to use “printilist” in this fashion.

```
$mkdir dir1

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 2      File name: filein
        Inode #: 3      File name: dir1

inode 2: plain file
is allocated

inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..

inode 4: plain file
unallocated
inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

“cpin examplefile /dir1/fileindir1” copies a file into the newly created directory. Program only uses absolute path names.

```
$cpin examplefile /dir1/fileindir1

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 2      File name: filein
        Inode #: 3      File name: dir1

inode 2: plain file
is allocated

inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 4      File name: fileindir1

inode 4: plain file
is allocated

inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

Next we issued “cpout” commands. The files copied out will be shown later after we remove files.

```
$cpout filein firstfileout

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$cpout /dir1/fileindir1 secondfileout

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$
```

Printilist before files are removed using “rm filename”

```
$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 2      File name: filein
        Inode #: 3      File name: dir1

inode 2: plain file
is allocated

inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 4      File name: fileindir1

inode 4: plain file
is allocated

inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

“rm filein”. Printilist now shows inode#2 is unallocated and directory no longer holds entry for file.

```
$rm filein

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 3      File name: dir1

inode 2: plain file
unallocated
inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 4      File name: fileindir1

inode 4: plain file
is allocated

inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

“rm /dir1/fileindir1” with the updated ilist.

```
$rm /dir1/fileindir1

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 3      File name: dir1

inode 2: plain file
unallocated
inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..

inode 4: plain file
unallocated
inode 5: plain file
unallocated
inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
unallocated
```

“mkdir /dir1/dir2”

```
$mkdir /dir1/dir2

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 3      File name: dir1

inode 2: plain file
unallocated
inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 5      File name: dir2

inode 4: plain file
unallocated
inode 5: directory
is allocated

Directory Entries for inode number # 5
        Inode #: 5      File name: .
        Inode #: 3      File name: ..

inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
```

“cpin examplefile /dir1/dir2/file4”

```
$cpin examplefile /dir1/dir2/file4

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 3      File name: dir1

inode 2: plain file
unallocated
inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 5      File name: dir2

inode 4: plain file
unallocated
inode 5: directory
is allocated

Directory Entries for inode number # 5
        Inode #: 5      File name: .
        Inode #: 3      File name: ..
        Inode #: 6      File name: file4

inode 6: plain file
is allocated

inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
```

“rm /dir1/dir2/file4”

```
$rm /dir1/dir2/file4

(a) mount filename
(b) ckfiletype inode #
(c) filesize inode #
(d) initfs fsize isize
(e) cpin filein newfilename
(f) cpout fileout newfilename
(g) mkdir newdirname
(h) rm filename
(i) printilist
(j) q


$printilist
inode 1: directory
is allocated

Directory Entries for inode number # 1
        Inode #: 1      File name: .
        Inode #: 1      File name: ..
        Inode #: 3      File name: dir1

inode 2: plain file
unallocated
inode 3: directory
is allocated

Directory Entries for inode number # 3
        Inode #: 3      File name: .
        Inode #: 1      File name: ..
        Inode #: 5      File name: dir2

inode 4: plain file
unallocated
inode 5: directory
is allocated

Directory Entries for inode number # 5
        Inode #: 5      File name: .
        Inode #: 3      File name: ..

inode 6: plain file
unallocated
inode 7: plain file
unallocated
inode 8: plain file
unallocated
inode 9: plain file
unallocated
inode 10: plain file
unallocated
inode 11: plain file
unallocated
inode 12: plain file
unallocated
inode 13: plain file
unallocated
inode 14: plain file
unallocated
inode 15: plain file
unallocated
inode 16: plain file
```

“q” for quit and then we concatenate the files we previously copied out from the v6 file system.

```
$q
Quit was selected
{cslinux2:~/newfolder/example} ls
examplefile  exfile  file.data  firstfileout  fsa  fsaccess.c  secondfileout
{cslinux2:~/newfolder/example} cat firstfileout
test
one
two
three
four
end
!!!!!!!

{cslinux2:~/newfolder/example} cat secondfileout
test
one
two
three
four
end
!!!!!!!

{cslinux2:~/newfolder/example}
```