Programming Assignment 3: Unix-like File System

Yuping Lu & Yue Tong

Unix-like File System File Structure:

...
…..Makefile
…..afs.cpp
…..afs.h
…..afs
…..output.txt
…..export.txt
…..sample.txt
…..samscript.sh
…..shell.h
…..virtual_disk.cpp

…..shell
…..…..cat.cpp
…..…..cd.cpp
…..…..close.cpp
…..…..cp.cpp
…..…..export.cpp
…..…..import.cpp
…..…..link.cpp
…..…..ls.cpp
…..…..mkdir.cpp
…..…..mkfs.cpp
…..…..open.cpp
…..…..read.cpp
…..…..rm.cpp
…..…..rmdir.cpp
…..…..seek.cpp
…..…..space.cpp
…..…..stat.cpp
…..…..tree.cpp
…..…..unlink.cpp
…..…..write.cpp

…..README.pdf

1. Compiling Source Code

The development environment we choose is Linux + C++.

Remove afs executable file if it exists, and then run the Makefile with command: make. We create a 128MB file named virtual_disk as a virtual disk for our file system. The virtual disk has 32768 blocks. And we use block 0 as the super block which contains the availability of other blocks using bitmap, block 1 as the root directory.

2. Implemented Shell Commands.

We implemented basic shell commands. These commands include mkfs, open, read, write, seek, close, mkdir, rmdir, cd, link, unlink, stat, ls, cat, cp, tree, import, export, home, space, exit.

1) mkfs
usage: Format the disk.
command: mkfs

2) open
usage: Create a file if it does not exist or open it with flag r, w, rw. It returns the fd.
command: open filename flag

3) read
usage: Read the content of a file with size. fd is needed. The file offset will move to current position.
command: read 3 10

4) write
usage: Write content to a file. The fd of the file is needed.
command: write 3 “Hello world”

5) seek
usage: Move the offset to a new position. fd is needed.
command: seek 3 10

6) close
usage: Close the file. fd is needed.
command: close 3

7) mkdir
usage: Create a sub directory under the current directory.
command:mkdir subdir
mkdir subdir/dir1

8) rmdir

usage: Remove the sub-directory and all the directories and files in it.
command: rmdir subdir

9) cd
usage: Change current directory.
command: cd subdir
cd ..

10) link
usage: Create a link named <dest> to an existing file named <src>
command: link src dest

11) unlink
usage: Remove a link of the file. If the file has no links, it will be removed.
command: unlink dest

12) stat
usage: Show the status of the file or directory with name <name>.
command: stat dest

13) ls
usage: Output the current subdirs and subfiles
command: ls

14) cat
usage: Output the given file
command: cat filename

15) cp
usage: Copy the file from a source to destination.
command: cp src dest

16) tree
usage: List the contents of the current directory in a tree-format. For each item listed, its type, date and file size are included.
command: tree

17) import
usage: Import a file from the host machine file system to the current directory.
command: import sample.txt sample

18) export
usage: Export a file from the current directory to the host machine file system.

command: export sample sample.txt

19) home
usage: Return to the root directory.
command: home

20) space
usage: Return the space left in the disk
command: space

21) exit
usage: Exit the file system
command: exit

3. Testing Files
samplescript.sh and sample.txt
We have changed samplescript.sh provided by Dr. Cao.
1) Because block 0 and 1 is used by our system, we changed the commands which need fd.
for example
write 0 "This is file1\n" to write 3 "This is file1\n".
The fd 3 is from your “open file1.txt w” operation
close 0 to close 3
read 0 5 to read 3 5
seek 0 5 t0 seek 3 5

2) We add export filelink export.txt to test the export command.

3) In the end we add home and tree command.

Note: You can use the samplescript.sh we provided, or your own’s. But you have to change the fd if you use your own’s test script.

sample.txt is a testing file for import command.

4. Running the application
./afs < samplescript.sh > output.txt
If you want run specific command, you can just start the application: ./afs

5. Output
The output is redirect to output.txt.
export.txt is the result of export command.