SNFS: Simple Network File System

Jetmir Berisha

How to run:
Run Makefile in the client and server directories.

Client: ./clientSNFS -port ### -address xxx -mount yyy -f

The -f is required. Order of command line arguments has to stay the same as above.

Server: ./serverSNFS -port ### -mount zzz

To close: ctrl-c the server first then the client.

Examples:

./clientSNFS -port 12344 -address localhost -mount /tmp/mntDir -f

./serverSNFS -port 12344 -mount /home/users/random/serverSNFS/mnt

Notes

1. Directories don't need to be created beforehand to qualify as mountpoints. The program will create the directories.

2. The path for mountpoint for the server should be an absolute path.

3. If you're using an already created directory and get a mounting error try unmounting that directory first with fusermount -u path. Then remove the directory and try again.

4. Multiple clients can connect and tamper with files but synchronization is not implemented on the server side. Each client is a separate thread.

This programming project taught me a lot about the workings of a file system. FUSE allows non-privileged users to create their own file systems without editing kernel code. This safe way of messing with filesystems was really entertaining and also enlightening at the same time. Not all aspects of this project were fun at first. Figuring out the basics for FUSE was daunting due to the lack of documentation.

The program starts with taking in the arguments in the following way.
Client: ./clientSNFS -port -address -mount -f

Server: ./serverSNFS -port -mount
These arguments in the program are parsed by getopt and put in the correct variables. For the client, the command line arguments are traversed once again so they can be added to a fuse_args struct. The fuse_args struct helps with providing fuse_main with a different set of commands separate from the command line arguments. This is necessary to do because fuse_main typically expects just a mount point. Then fuse_main is called with `fuse_main(args. argc, args.argv,`

`&operations, NULL);` where args is a fuse_args struct, args.argc is the count of the arguments and args.argv contains the mount point and '-f' if provided. When fuse_main is called the mountpoint is changed into a fuse mounted directory and it stays like that until the user executes a ctrl-c or call fusermount -u on the mounted path.

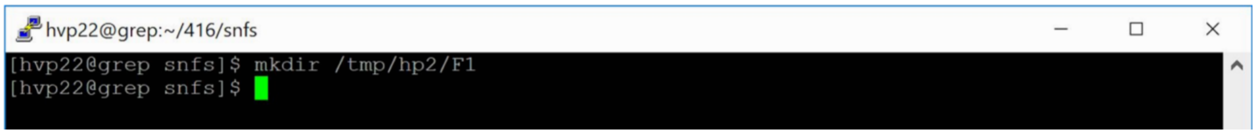
The client and server protocol is as follows. Assuming the initial connection is made between the client and the server, the server assigns a thread for each client. In the thread function the server is always waiting on commands. For each callback function on the client side, it would at the appropriate time send a command to the server describing the function. So, for getattr the client would send the "sf_getattr" command. The server waits in a loop for commands. The server after receiving a command goes to the matching section. The client then sends the appropriate data to the server which it is already expecting. For each callback function the client creates a new socket to send data on. After doing the necessary computations and system calls the server returns the computed data. The client fills in the data (if required) or parses the server's response and passes it to FUSE.

Examples on CentOS:

Scenario: After running both makefiles have 3 terminals open. One runs the server, another runs the client, and the last one is opened after the former in the mounted directory on the client side.

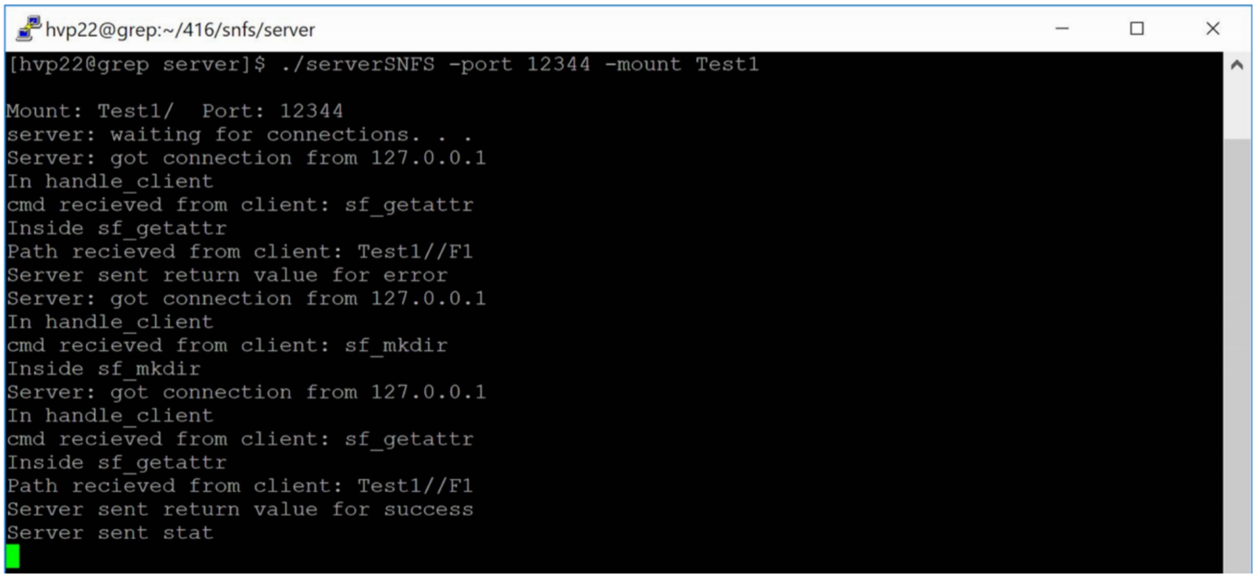
Starting with a fresh mount point on server side called Test1.

1. Mkdir: call mkdir path. Proves getattr & mkdir. Command:


hvp22@grep:~/416/snfs

```
[hvp22@grep snfs]$ mkdir /tmp/hp2/F1
[hvp22@grep snfs]$
```

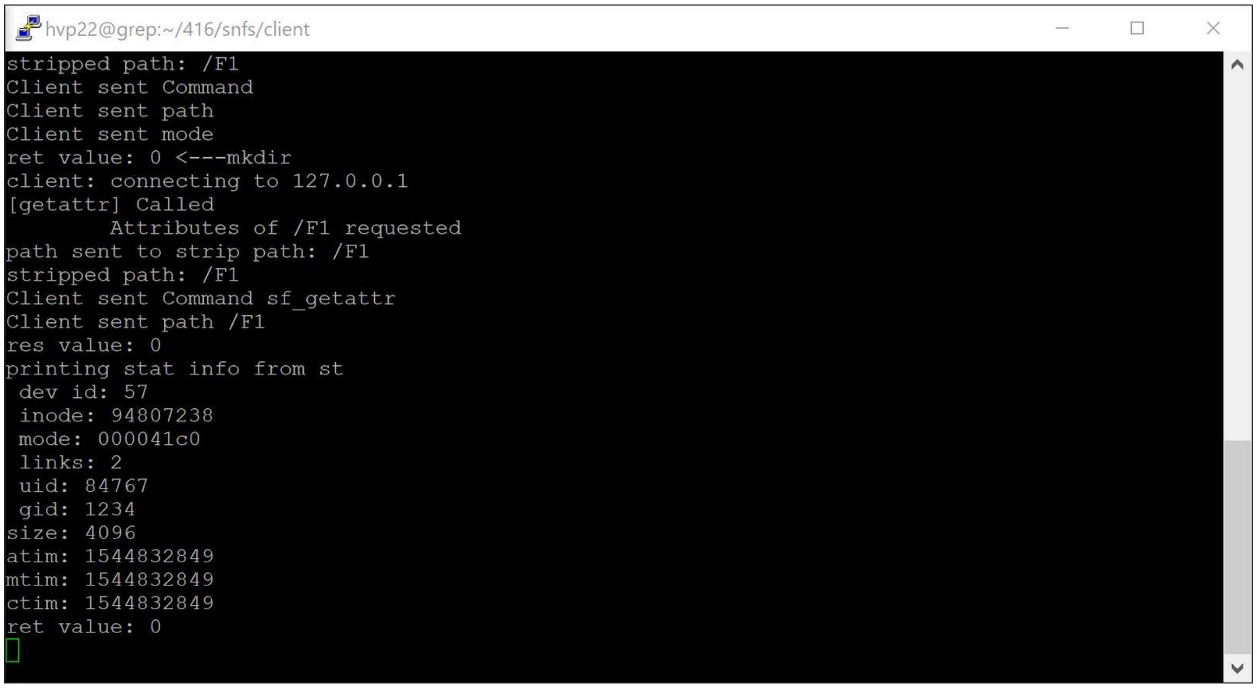
Server:


hvp22@grep:~/416/snfs/server

```
[hvp22@grep server]$ ./serverSNFS -port 12344 -mount Test1

Mount: Test1/  Port: 12344
server: waiting for connections. . .
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_getattr
Inside sf_getattr
Path recieved from client: Test1//F1
Server sent return value for error
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_mkdir
Inside sf_mkdir
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_getattr
Inside sf_getattr
Path recieved from client: Test1//F1
Server sent return value for success
Server sent stat
```

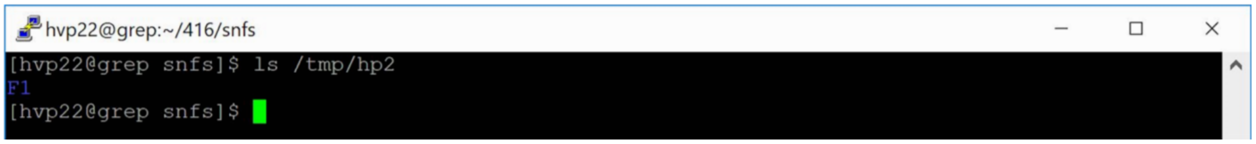
Client:



hvp22@grep:~/416/snfs/client

```
stripped path: /F1
Client sent Command
Client sent path
Client sent mode
ret value: 0 <---mkdir
client: connecting to 127.0.0.1
[getattr] Called
        Attributes of /F1 requested
path sent to strip path: /F1
stripped path: /F1
Client sent Command sf_getattr
Client sent path /F1
res value: 0
printing stat info from st
 dev id: 57
 inode: 94807238
 mode: 000041c0
 links: 2
 uid: 84767
 gid: 1234
size: 4096
atim: 1544832849
mtim: 1544832849
ctim: 1544832849
ret value: 0
```

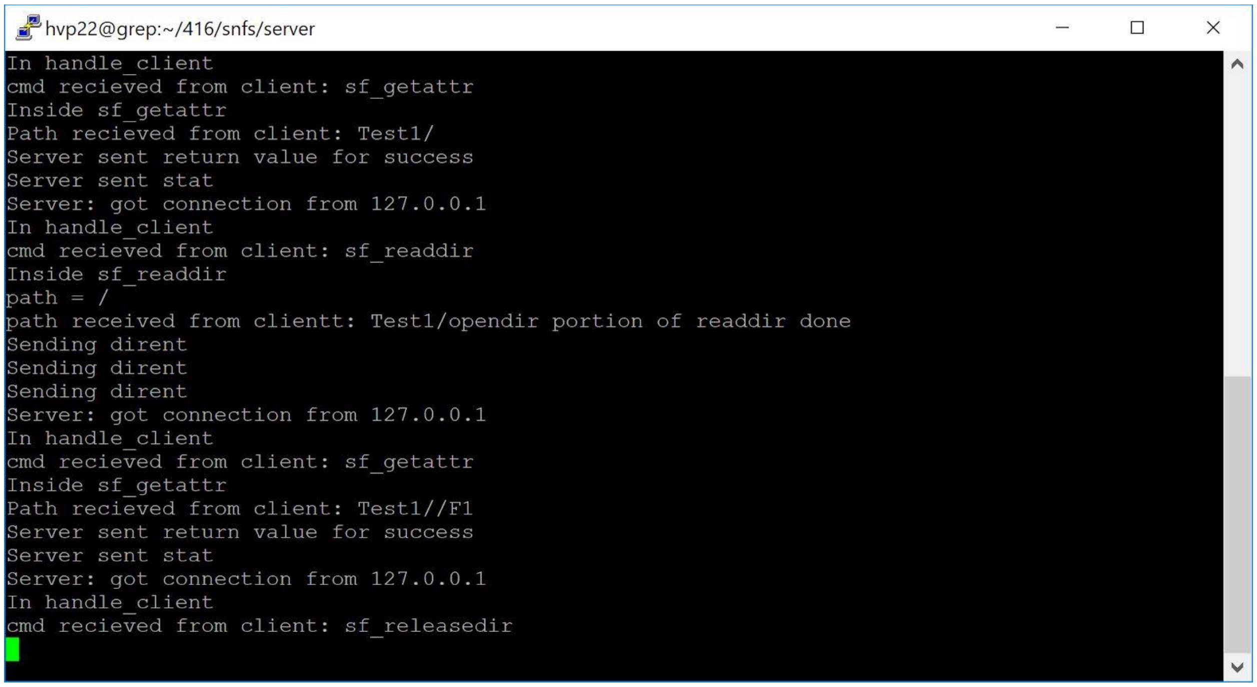
2. **ls path**. Proves getattr & readdir & opendir & releasedir. Command:



hvp22@grep:~/416/snfs

```
[hvp22@grep snfs]$ ls /tmp/hp2
F1
[hvp22@grep snfs]$
```

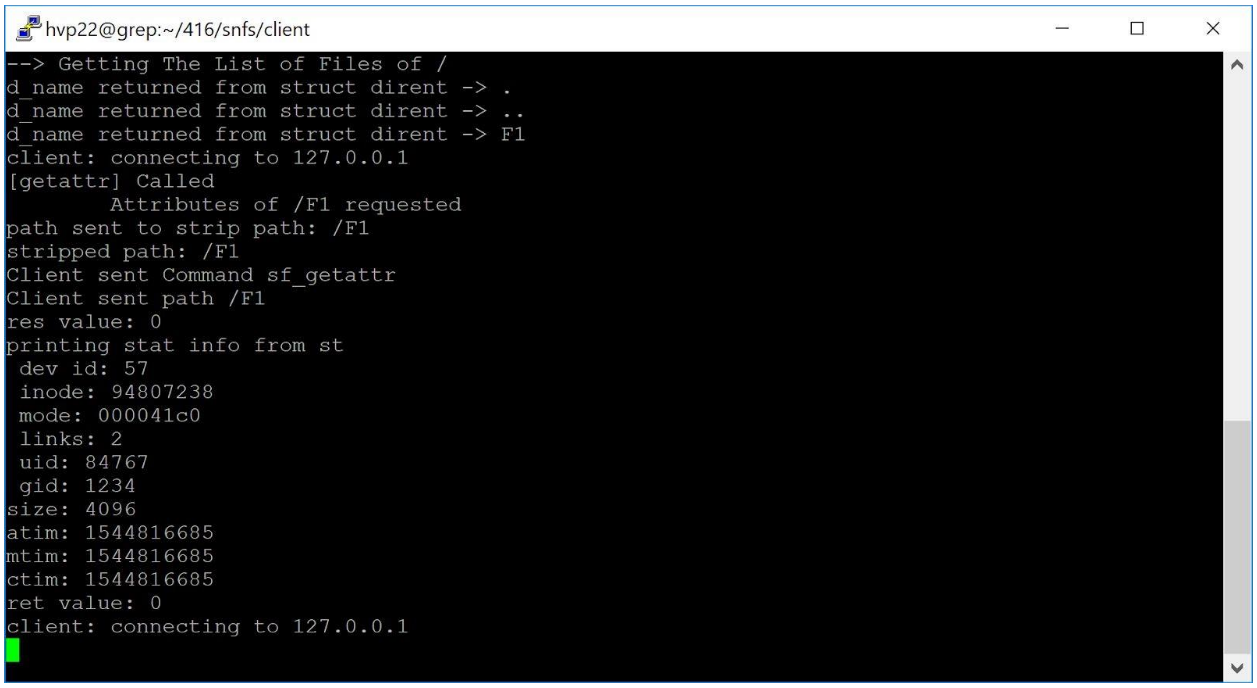
Server:



hvp22@grep:~/416/snfs/server

```
In handle_client
cmd recieved from client: sf_getattr
Inside sf_getattr
Path recieved from client: Test1/
Server sent return value for success
Server sent stat
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_readdir
Inside sf_readdir
path = /
path received from clientt: Test1/opendir portion of readdir done
Sending dirent
Sending dirent
Sending dirent
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_getattr
Inside sf_getattr
Path recieved from client: Test1//F1
Server sent return value for success
Server sent stat
Server: got connection from 127.0.0.1
In handle_client
cmd recieved from client: sf_releasedir
```

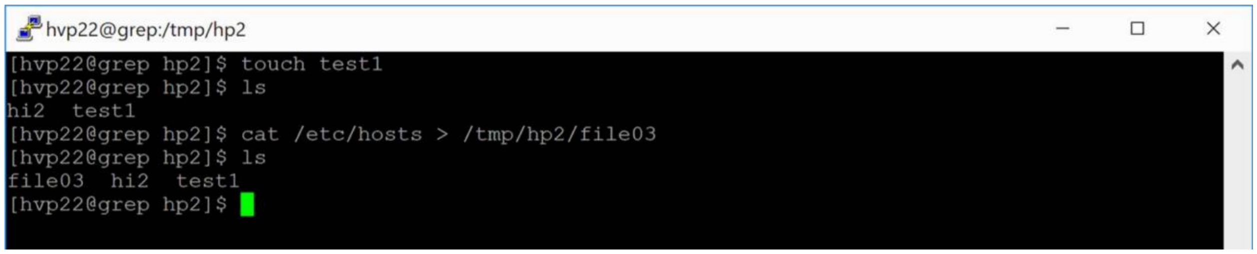
Client:



hvp22@grep:~/416/snfs/client

```
--> Getting The List of Files of /
d_name returned from struct dirent -> .
d_name returned from struct dirent -> ..
d_name returned from struct dirent -> F1
client: connecting to 127.0.0.1
[getattr] Called
        Attributes of /F1 requested
path sent to strip path: /F1
stripped path: /F1
Client sent Command sf_getattr
Client sent path /F1
res value: 0
printing stat info from st
 dev id: 57
 inode: 94807238
 mode: 000041c0
 links: 2
 uid: 84767
 gid: 1234
size: 4096
atim: 1544816685
mtim: 1544816685
ctim: 1544816685
ret value: 0
client: connecting to 127.0.0.1
```

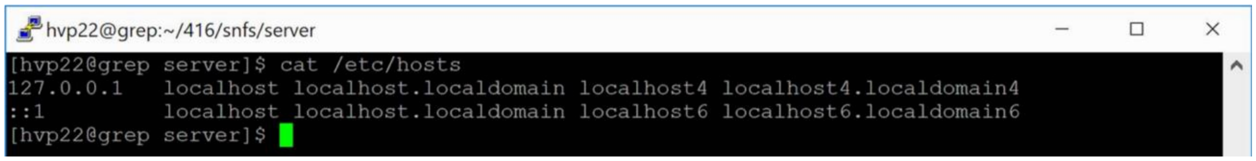
3. cat /etc/hosts > /tmp/hp2/file03   Proves write & read & open & create & release.
   Command:



hvp22@grep:/tmp/hp2

```
[hvp22@grep hp2]$ touch test1
[hvp22@grep hp2]$ ls
hi2  test1
[hvp22@grep hp2]$ cat /etc/hosts > /tmp/hp2/file03
[hvp22@grep hp2]$ ls
file03  hi2  test1
[hvp22@grep hp2]$
```

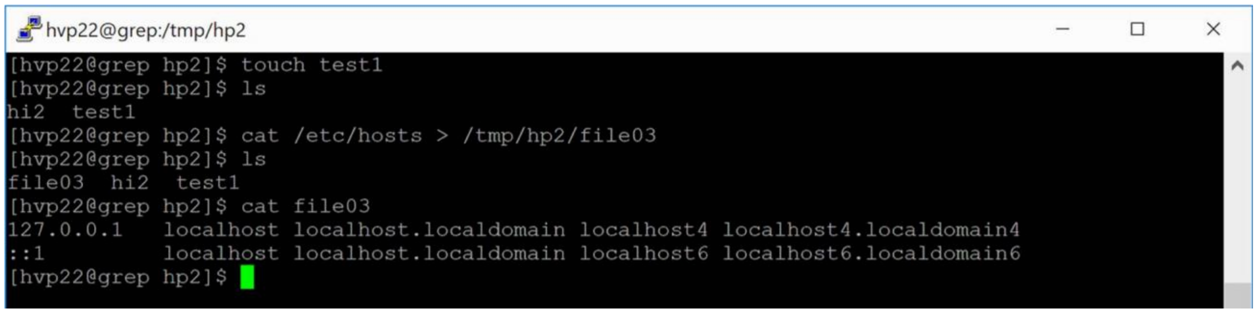
Original content of Hosts:



hvp22@grep:~/416/snfs/server

```
[hvp22@grep server]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[hvp22@grep server]$
```

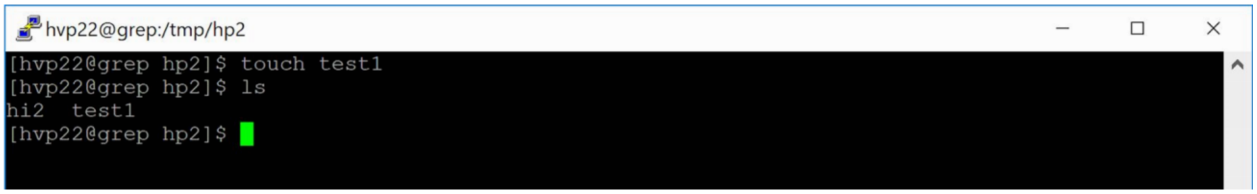
New file created content:



hvp22@grep:/tmp/hp2

```
[hvp22@grep hp2]$ touch test1
[hvp22@grep hp2]$ ls
hi2  test1
[hvp22@grep hp2]$ cat /etc/hosts > /tmp/hp2/file03
[hvp22@grep hp2]$ ls
file03  hi2  test1
[hvp22@grep hp2]$ cat file03
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[hvp22@grep hp2]$
```

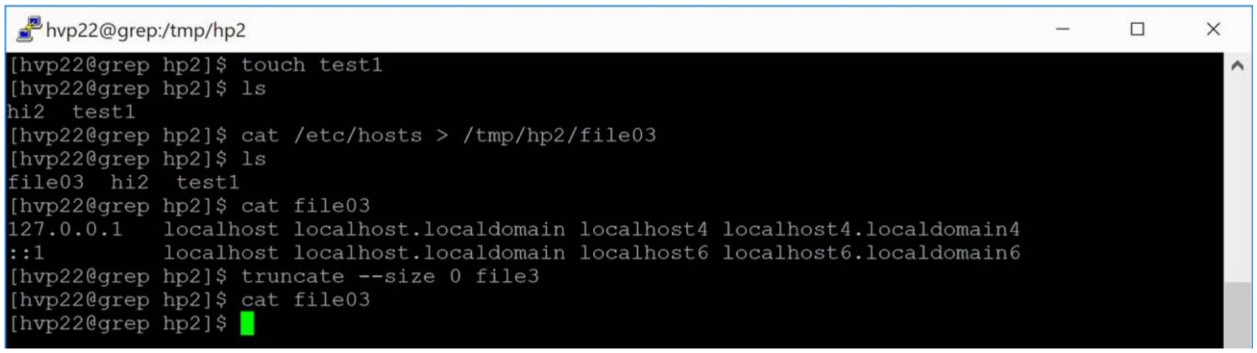
4. Touch:



hvp22@grep:/tmp/hp2

```
[hvp22@grep hp2]$ touch test1
[hvp22@grep hp2]$ ls
hi2  test1
[hvp22@grep hp2]$ 
```

5. Truncate:



hvp22@grep:/tmp/hp2

```
[hvp22@grep hp2]$ touch test1
[hvp22@grep hp2]$ ls
hi2  test1
[hvp22@grep hp2]$ cat /etc/hosts > /tmp/hp2/file03
[hvp22@grep hp2]$ ls
file03  hi2  test1
[hvp22@grep hp2]$ cat file03
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[hvp22@grep hp2]$ truncate --size 0 file3
[hvp22@grep hp2]$ cat file03
[hvp22@grep hp2]$ 
```