

CS450-PA4

Jenifer Rodriguez Delgado A20398938

Yousef Suleiman A20463895

Test Data

Equivalence Partitioning	
1 Unreachable empty directory	2 Unreachable file
3 Multiple unreachable empty directories	4 Multiple unreachable files
5 Tree of unreachable directories	
6 Multiple Trees	

1. Unreachable empty directory

```
$ mkdir d d/a
$ directoryWalker -h
. 1 1 512
./README 2 2 2286
./cat 2 3 16392
./echo 2 4 15244
./forktest 2 5 9548
./grep 2 6 18612
./init 2 7 15832
./kill 2 8 15272
./ln 2 9 15124
./ls 2 10 17764
./mkdir 2 11 15372
./rm 2 12 15348
./sh 2 13 27984
./stressfs 2 14 16260
./usertests 2 15 67368
./wc 2 16 17124
./zombie 2 17 14940
./directoryWalke 2 18 19084
./inodeTBWalker 2 19 17300
./Walkers 2 20 21588
./corruptor 2 21 16256
./console 3 22 0
./d 1 23 48
./d/a 1 24 32

$ corruptor d
d 1 23 48
$ directoryWalker -h
. 1 1 512
./README 2 2 2286
./cat 2 3 16392
./echo 2 4 15244
./forktest 2 5 9548
./grep 2 6 18612
./init 2 7 15832
./kill 2 8 15272
./ln 2 9 15124
./ls 2 10 17764
./mkdir 2 11 15372
./rm 2 12 15348
./sh 2 13 27984
./stressfs 2 14 16260
./usertests 2 15 67368
./wc 2 16 17124
./zombie 2 17 14940
./directoryWalke 2 18 19084
./inodeTBWalker 2 19 17300
./Walkers 2 20 21588
./corruptor 2 21 16256
./console 3 22 0
$
```

A directory `d` is made holding a single directory `a`. Program `directoryWalker` can see them. However, after running `corruptor` on `d`, directory `a` is no longer reachable.

```

$ Walkers
Number of unreachable files: 1
1 24 32
$ directoryWalker -h
. 1 1 512
./README 2 2 2286
./cat 2 3 16392
./echo 2 4 15244
./forktest 2 5 9548
./grep 2 6 18612
./init 2 7 15832
./kill 2 8 15272
./ln 2 9 15124
./ls 2 10 17764
./mkdir 2 11 15372
./rm 2 12 15348
./sh 2 13 27984
./stressfs 2 14 16260
./usertests 2 15 67368
./wc 2 16 17124
./zombie 2 17 14940
./directoryWalke 2 18 19084
./inodeTBWalker 2 19 17300
./Walkers 2 20 21588
./corruptor 2 21 16256
./console 3 22 0
./lost+found-24 1 24 32
$
$ cd lost+found-24
$ /ls
. 1 24 32
.. 1 1 512
$ cd .
$ cd ..
$ ls
. 1 1 512
.. 1 1 512
README 2 2 2286
cat 2 3 16392
echo 2 4 15244
forktest 2 5 9548
grep 2 6 18612
init 2 7 15832
kill 2 8 15272
ln 2 9 15124
ls 2 10 17764
mkdir 2 11 15372
rm 2 12 15348
sh 2 13 27984
stressfs 2 14 16260
usertests 2 15 67368
wc 2 16 17124
zombie 2 17 14940
directoryWalke 2 18 19084
inodeTBWalker 2 19 17300
Walkers 2 20 21588

```

After running `Walkers`, the hanging directory was found and placed in the root as `lost+found-24` and it could be entered and left with `.` and `..` working properly.

2. Unreachable file

```

$ mkdir d
$ echo foo > d/a
$ corruptor d
d 1 23 48
$ Walkers
Number of unreachable files: 1
2 24 4
$ cat lost+found-24
foo
$ 

```

A similar scenario as before but now a very important file is placed in directory `d` before it is corrupted. `Walkers` recovers it as `lost+found-24` and we can still read its contents.

3. Multiple unreachable empty directories

```
$ mkdir d d/a d/b d/c
$ corruptor d
d 1 23 80
$ Walkers
Number of unreachable files: 3
1 24 32
1 25 32
1 26 32
$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
```

We make multiple directories in `d` and corrupt `d` again. Now `a`, `b` and `c` are all hanging. After calling `Walkers`, all the hanging directories are found and placed into the root directory.

```
inodeTBWalker 2 19 17300
Walkers        2 20 21588
corruptor      2 21 16256
console        3 22 0
lost+found-24  1 24 32
lost+found-25  1 25 32
lost+found-26  1 26 32
$
```

4. Multiple unreachable files

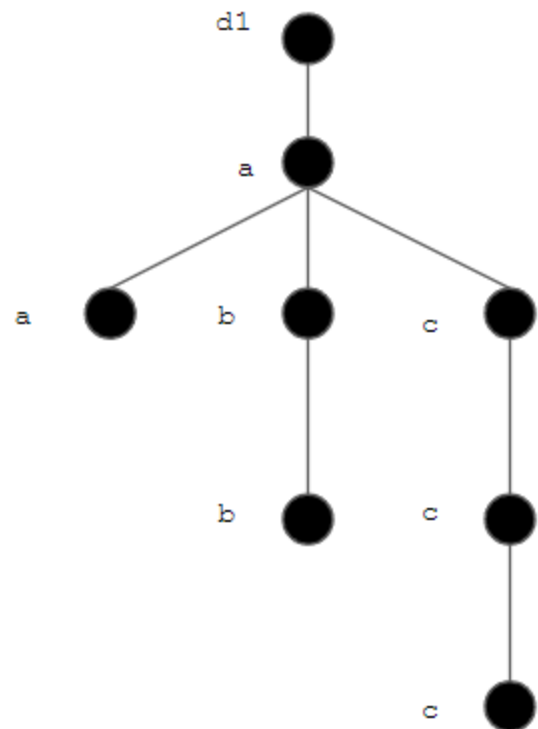
```
init: starting sh
$ mkdir d
$ echo foo > d/a
$ echo bar > d/b
$ echo cat > d/c
$ corruptor d
d 1 23 80
$ Walkers
Number of unreachable files: 3
2 24 4
2 25 4
2 26 4
$ cat lost+found-24
foo
$ cat lost+found-25
bar
$ cat lost+found-26
cat
```

We place multiple files in `d`, corrupt it, use `Walkers` to salvage them, and demonstrate that they are still readable.

5 Tree of unreachable directories

```
$ mkdir d1 d1/a d1/a/a d1/a/b d1/a/c
$ mkdir d1/a/b/b d1/a/c/c d1/a/c/c/c
$ directoryWalker -h
. 1 1 512
./README 2 2 2286
./cat 2 3 16392
```

```
./d1 1 23 48
./d1/a 1 24 80
./d1/a/a 1 25 32
./d1/a/b 1 26 48
./d1/a/b/b 1 28 32
./d1/a/c 1 27 48
./d1/a/c/c 1 29 48
./d1/a/c/c/c 1 30 32
$
```



We make a tree of directories that looks as such. Next we corrupt d1. So d/a and everything under it is unreachable. However after calling Walkers:

```
corruptor: cannot open d
$ corruptor d1
d1 1 23 48
$ Walkers
Number of unreachable files: 7
1 24 80
1 25 32
1 26 48
1 27 48
1 28 32
1 29 48
1 30 32
```

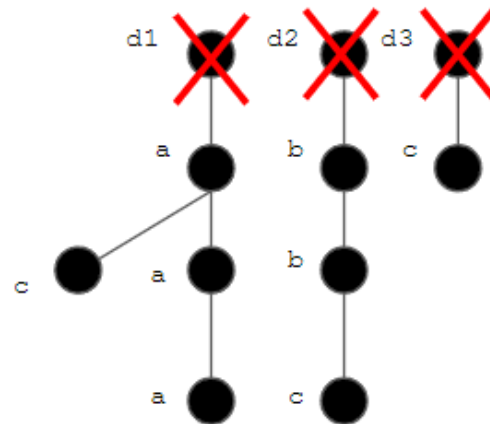
```
WC          2 16 17124
zombie      2 17 14940
directoryWalke 2 18 19084
inodeTBWalker 2 19 17300
Walkers     2 20 21588
corruptor   2 21 16256
console     3 22 0
lost+found-24 1 24 80
$
```

we can see that it found all 7 hanging directories, but when calling ls, it only places one of the nodes in the root directory as the other nodes resided under it. We can see this in directoryWalker

```
./wc 2 16 17124
./zombie 2 17 14940
./directoryWalke 2 18 19084
./inodeTBWalker 2 19 17300
./Walkers 2 20 21588
./corruptor 2 21 16256
./console 3 22 0
./lost+found-24 1 24 80
./lost+found-24/a 1 25 32
./lost+found-24/b 1 26 48
./lost+found-24/b/b 1 28 32
./lost+found-24/c 1 27 48
./lost+found-24/c/c 1 29 48
./lost+found-24/c/c/c 1 30 32
```

6. Multiple Trees

```
init: starting sh
$ mkdir d1 d1/a d1/a/a d1/a/a/a
$ mkdir d2 d2/b d2/b/b
$ mkdir d3
$ echo foo > d3/c
$ echo bar > d1/a/c
$ echo cat > d2/b/b/c
$ corruptor d1 d2 d3
d1 1 23 48
d2 1 27 48
d3 1 30 48
$ Walkers
Number of unreachable files: 8
1 24 64
1 25 48
1 26 32
1 28 48
1 29 48
2 31 4
2 32 4
2 33 4
$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
cat       2 3 16392
echo      2 4 15244
forktest  2 5 9548
grep      2 6 18612
init      2 7 15832
kill      2 8 15272
ln        2 9 15124
ls        2 10 17764
mkdir     2 11 15372
rm        2 12 15348
sh        2 13 27984
stressfs  2 14 16260
usertests 2 15 67368
wc        2 16 17124
zombie    2 17 14940
directoryWalker 2 18 19084
inodeTBWalker 2 19 17300
Walkers   2 20 21588
corruptor 2 21 16256
console   3 22 0
lost+found-24 1 24 64
lost+found-28 1 28 48
lost+found-31 2 31 4
$
```



This time we make 3 different trees starting with d1 d2 and d2 then place some c files in them. We corrupt d1 d2 and d2 and their children files become unreachable. After running Walkers, it places the files in 3 lost+found directories. We demonstrate that the c files are still readable.

```
./lost+found-24 1 24 64
./lost+found-24/a 1 25 48
./lost+found-24/a/a 1 26 32
./lost+found-24/c 2 32 4
./lost+found-28 1 28 48
./lost+found-28/b 1 29 48
./lost+found-28/b/c 2 33 4
./lost+found-31 2 31 4
$ cat lost+found-24/c
bar
$ cat lost+found-28/b/c
cat
$ cat lost+found-31
foo
$
```