PSET 2

Jonathan Lam
Prof. Hakner
ECE 357
Operating Systems
9/25/19

Problem 1: A simple hypothetical filesystem

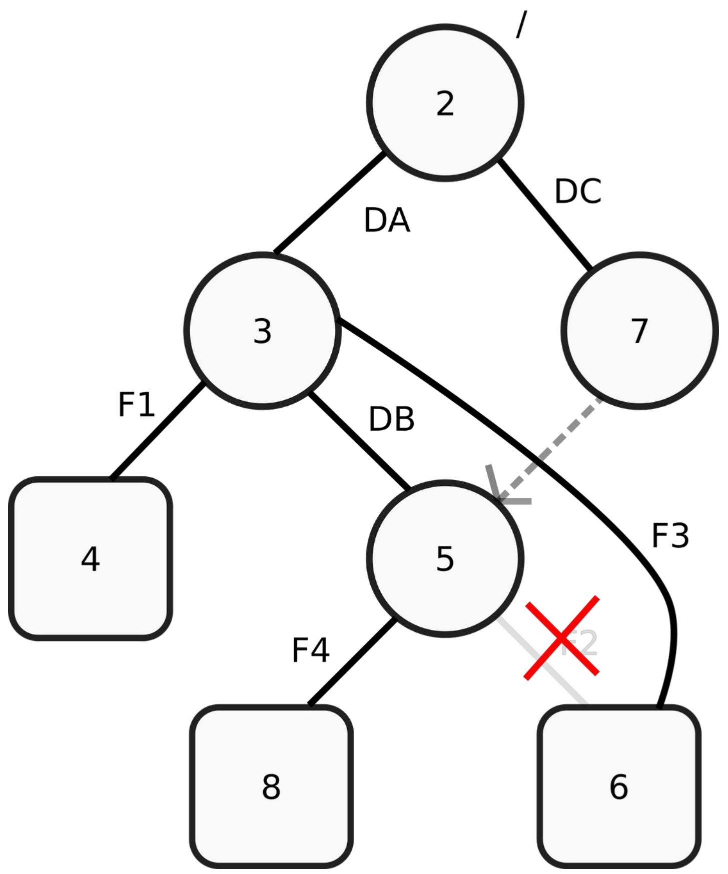
Inode tables:

#	type	nlink	direntry table (if dir)	
			inode	name
2	dir	3	2	.
			2	..
			3	DA
			7	DC
3	dir	3	3	.
			2	..
			4	F1
			5	DB
			6	F3
4	file	1	–	
5	dir	2	5	.
			3	..
			8	F4
6	file	1	–	
7	symlink	1	–	
8	file	1	–	

PSET 2

Jonathan Lam
Prof. Hakner
ECE 357
Operating Systems
9/25/19

Filesystem tree (/ is mounted at /mnt of root fs)



Problem 2: Exploratory questions

A) Data corruption without journaling

<u>What program will need to be run before the volume can be mounted?</u>
fsck is the command to check for data corruption (inconsistencies on the disk).

<u>What sort of issues do we expect this program will find with the volume? Give a specific example of at least one issue.</u>
An example of data corruption fsck can notice is incomplete file deletion or insertion caused by the abrupt power off. What fsck does is recursively traverse the filesystem tree from the root, checking inode and free block maps, and then checking for orphaned inodes and free blocks. For example, during a file creation, if power was cut after creating an inode but before it is inserted into a directory's data block, fsck will detect the orphaned inode. Similarly, during file deletion, if the inode is deleted before the directory entry is removed (or vice

versa, depending on the implementation of remove), then fsck should find this inconsistency as well.

<u>If this is a 2TB volume with 1,000,000 allocated inodes, will it take a long time? Why or why not?</u>
Given that fsck has to traverse the entire filesystem tree and scan the entire inode table in order to check for inconsistencies. Since fsck has to go through the data blocks starting from root and check associated inodes, and it has to scan through each data block (~2TB/block size, a very large number) to make sure that free data blocks are not claimed by any files, this will take a long time.

B) EACCESS when deleting /dir/foo
If the user doesn't have write permissions to dir, then deleting the file (i.e., changing the directory name-inode table) is not allowed. Also, if the user doesn't have execute/traverse permissions to dir or root, then they cannot traverse root (/) or dir (/dir) (and therefore do not have permissions to do any action on foo since it traverses dir.

C) Possible factors:
<u>Metadata/indirect blocks</u>: The size of a volume includes more than only file contents (i.e., data blocks): it also contains the other parts of the unix filesystem, such as the inode table and indirect blocks in ext3 (or extend descriptors in ext4 data blocks).

<u>Definition of 4TB (marketing scamminess)</u>: this can be a base 10 number (4*10^12) or a base 2 number (2^42). 4000 videos that are each 2^30 bytes long would have a total size (in data blocks alone) of 4.29*10^12 bytes, which is significantly larger than the base 10 version, while the base 2 version has 4.39*10^12 bytes (which would be sufficient and was what the user was looking for).

<u>Reserve factor (possibly)</u>: for performance, it helps to leave free blocks abundant so files can be extended into adjacent free blocks (to avoid more disk activity and therefore slower times in mechanical hard drives), i.e., to reduce fragmentation. This may mean leaving some space that cannot be allocated, therefore effectively shrinking hard drive size from the actual capacity.

D) Causes of different move speeds
It is possible that F1 and F2 are on the same volume, but F2 and F3 are not on the same volume. This is possible because it is possible that Z is mounted to a different volume (but looks like an ordinary directory because of the VFS), and therefore the mv command copies the files instead of linking/unlinking (since hard links cannot be made across volumes). Copying a large file is much slower than linking/unlinking.

PSET 2

Jonathan Lam
Prof. Hakner
ECE 357
Operating Systems
10/2/19

PROGRAM SOURCE (rls.c)
```
#include <ctype.h>
#include <dirent.h>
#include <errno.h>
#include <grp.h>
#include <pwd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/sysmacros.h>
#include <sys/types.h>
#include <time.h>
#include <unistd.h>

// logging to stderr
#define ERROR_NON_FATAL(action, ctx, msg) \
  fprintf(stderr, "rls: %s \"%s\": %s\n", action, ctx, msg);

#define ERROR_FATAL(action, ctx, msg) {\
    fprintf(stderr, "rls: ERROR: %s \"%s\": %s\n", action, ctx, msg);\
    exit(EXIT_FAILURE);\
  }

// conditions for printing; used twice hence the macro
#define should_print()\
      (!optm || ((optm > 0 && stime - f_stat.st_mtime > optm)\
             || (optm < 0 && stime - f_stat.st_mtime <= -optm)))\
       && rperm(&f_stat, u_info)

// struct to store user/group info for optu
struct u_info {
  int optu;
  uid_t uid;
  gid_t *groups;
  int ngroups;
};

// get whether or not a user has read perms
int rperm(struct stat *f_stat, struct u_info *u_info) {
  // if optu not set or root, ignore
  if(!u_info->optu || !u_info->uid)
    return 1;

  // replicate unix perm checking; u->g->o
  if(f_stat->st_uid == u_info->uid)
    return f_stat->st_mode & S_IRUSR;
  for(int i = 0; i < u_info->ngroups; i++)
    if(u_info->groups[i] == f_stat->st_gid)
      return f_stat->st_mode & S_IRGRP;
  if(f_stat->st_mode & S_IROTH)
    return 1;
  return 0;
}

// print directory entry; handles decoding of stat info
void print_dirent(struct stat *f_stat, char *f_path) {
  // maximum path length in ext4: https://unix.stackexchange.com/a/32834/307410
  // maximum username length: https://serverfault.com/a/294122/332775
  char f_uname[33], f_gname[32], f_size[32], rl_buf[4097], f_link[4101],
       f_mtime[16], f_mode[11];
  struct passwd *f_user;
  struct group *f_group;
  int rl_len, f_blocks;

  // decode mode
  // get file type
  switch(f_stat->st_mode & S_IFMT) {
    case S_IFBLK: *f_mode = 'b'; break;
    case S_IFCHR: *f_mode = 'c'; break;
    case S_IFDIR: *f_mode = 'd'; break;
    case S_IFIFO: *f_mode = 'p'; break;
    case S_IFLNK: *f_mode = 'l'; break;
    case S_IFSOCK:*f_mode = 's'; break;
    case S_IFREG: *f_mode = '-'; break;
    default:      *f_mode = '?';
  }

  // permission bits
  f_mode[1] = f_stat->st_mode & S_IRUSR ? 'r' : '-';
  f_mode[2] = f_stat->st_mode & S_IWUSR ? 'w' : '-';
  f_mode[3] = f_stat->st_mode & S_IXUSR
            ? f_stat->st_mode & S_ISUID ? 's' : 'x'
            : f_stat->st_mode & S_ISUID ? 'S' : '-';
  f_mode[4] = f_stat->st_mode & S_IRGRP ? 'r' : '-';
  f_mode[5] = f_stat->st_mode & S_IWGRP ? 'w' : '-';
  f_mode[6] = f_stat->st_mode & S_IXGRP
            ? f_stat->st_mode & S_ISGID ? 's' : 'x'
            : f_stat->st_mode & S_ISGID ? 'S' : '-';
  f_mode[7] = f_stat->st_mode & S_IROTH ? 'r' : '-';
  f_mode[8] = f_stat->st_mode & S_IWOTH ? 'w' : '-';
  f_mode[9] = f_stat->st_mode & S_IXOTH
            ? f_stat->st_mode & S_ISVTX ? 't' : 'x'
            : f_stat->st_mode & S_ISVTX ? 'T' : '-';
  f_mode[10] = '\0';

  // get user info
  if(f_user = getpwuid(f_stat->st_uid))
    strcpy(f_uname, f_user->pw_name);
  else
    sprintf(f_uname, "%d", f_stat->st_uid);

  // get group info
  if(f_group = getgrgid(f_stat->st_gid))
    strcpy(f_gname, f_group->gr_name);
  else
    sprintf(f_gname, "%d", f_stat->st_gid);

  // if not char/block dev, get size in bytes
  // else get dev major, minor number
  if(!S_ISBLK(f_stat->st_mode) && !S_ISCHR(f_stat->st_mode))
    sprintf(f_size, "%ld", f_stat->st_size);
  else
    sprintf(f_size, "%d,%d", major(f_stat->st_rdev), minor(f_stat->st_rdev));

  // format date; if within the last year, do date/time (mon day time)
  // if longer do date/year (mon day year)
  // (1 year = 365*24*60*60 seconds = 31536000s)
  if(time(NULL) - f_stat->st_mtime < 31536000)
    strftime(f_mtime, 16, "%b %e %H:%M", localtime(&f_stat->st_mtime));
  else
    strftime(f_mtime, 16, "%b %e  %Y", localtime(&f_stat->st_mtime));

  // if symlink, get what it points to; otherwise, clear string
  if(S_ISLNK(f_stat->st_mode)) {
    if((rl_len = readlink(f_path, rl_buf, sizeof rl_buf - 1)) > 0)
      rl_buf[rl_len] = '\0';
    else
      ERROR_NON_FATAL("readlink", f_path, strerror(errno));

    sprintf(f_link, " -> %s", rl_buf);
  } else
    *f_link = '\0';

  // get number of blocks; contribution by Dave Kwong in the case
  // that fs has weird blocksize (adds 1 if not even multiple of 512 bytes)
  f_blocks = f_stat->st_blocks/2 + f_stat->st_blocks%2;

  fprintf(stdout, " %8ld %6d %s %3d %-8s %-8s %8s %s %s%s\n",
          f_stat->st_ino, f_blocks, f_mode, f_stat->st_nlink,
          f_uname, f_gname, f_size, f_mtime, f_path, f_link);
  return;
};

// recursively walk fs and print; handles tree walking and deciding
// which inode entries should be printed
void rls(char *basedir, int optm, int optv,
         dev_t start_vol, struct u_info *u_info, time_t stime) {
  DIR *dir;
  struct dirent *dirent;
  // max path length is 4096: https://unix.stackexchange.com/a/32834/307410
  char f_path[4097];
  struct stat f_stat;

  if(!(dir = opendir(basedir)))
    // if no access to directory, exit here
    if(errno == EACCES) {
      ERROR_NON_FATAL("opening directory", basedir, strerror(errno));
      return;
    } else
      ERROR_FATAL("opening directory", basedir, strerror(errno));

  // this happens on the first call to rls (by main() driver)
  if(start_vol == -1) {
    if(stat(basedir, &f_stat) < 0)
      ERROR_FATAL("stat directory", basedir, strerror(errno));
    start_vol = f_stat.st_dev;

    // print if applicable
    if(should_print())
      print_dirent(&f_stat, basedir);
  }

  errno = 0;
  while(dirent = readdir(dir)) {
    // ignore ., ..
    if(!strcmp(dirent->d_name, ".") || !strcmp(dirent->d_name, ".."))
      continue;

    // stat inode
    sprintf(f_path, "%s%s%s", basedir,
            basedir[strlen(basedir)-1] == '/' ? "" : "/", dirent->d_name);
    if(lstat(f_path, &f_stat) < 0) {
      ERROR_NON_FATAL("stat", basedir, strerror(errno));
      if(errno == EACCES)
        return;
    }

    // print if applicable
    if(should_print())
      print_dirent(&f_stat, f_path);
    errno = 0;

    // if directory, recursively print filenames
    if(S_ISDIR(f_stat.st_mode))
      if(optv && f_stat.st_dev != start_vol) {
        ERROR_NON_FATAL("mount point", f_path, "Not crossing mount point");
      } else
        rls(f_path, optm, optv, start_vol, u_info, stime);
    errno = 0;
  }
  // error reading directory in above loop
  if(errno)
    ERROR_NON_FATAL("reading directory", basedir, strerror(errno));

  // close dir
  if(closedir(dir) < 0)
    ERROR_FATAL("closing directory", basedir, strerror(errno));

  return;
}

// driver for rls function
int main(int argc, char **argv) {
  int optv = 0, optm = 0, opt;
  long optu = -1;
  char *startdir = "./";
  struct passwd *optu_passwd;
  struct u_info u_info = { .optu = 0 };
  time_t stime;

  // parse args
  while((opt = getopt(argc, argv, "m:u:v")) != -1) {
    switch(opt) {
      case 'm':
        // silently fail if invalid number; atoi will return 0 if invalid
        optm = atoi(optarg);
        break;
      case 'u':
        errno = 0;
        if((isdigit(*optarg) && (optu_passwd = getpwuid(atoi(optarg))))
          || (!isdigit(*optarg) && (optu_passwd = getpwnam(optarg)))) {
          // get grouplist; once to get ngroups, second time to retrieve gl
          u_info.ngroups = 0;
          getgrouplist(optu_passwd->pw_name, optu_passwd->pw_gid,
                       NULL, &u_info.ngroups);
          u_info.groups = (gid_t *) malloc(u_info.ngroups * sizeof(gid_t *));
          getgrouplist(optu_passwd->pw_name, optu_passwd->pw_gid,
                       u_info.groups, &u_info.ngroups);
          u_info.optu = 1;
          u_info.uid = optu_passwd->pw_uid;
        }
        if(errno)
          ERROR_FATAL("(getpwuid/getpwnam) processing parameter -u", optarg,
                      strerror(errno));

        // user not found; non-fatal error
        if(!optu_passwd)
          ERROR_NON_FATAL("processing parameter -u", optarg, "User not found");

        break;
      case 'v':
        optv = 1;
        break;
    }
  }

  // iterate thu args if given; else rls on cwd
  stime = time(NULL);
  if(optind == argc)
    rls(".", optm, optv, -1, &u_info, stime);
  else
    while(optind++ != argc)
      rls(argv[optind-1], optm, optv, -1, &u_info, stime);

  // free dynamically allocated memory and exit
  if(u_info.optu)
    free(u_info.groups);
  exit(EXIT_SUCCESS);
}
/**
  * Notes about this program:
  * - Formatting (padding) is not dynamic, but not the point of this project.
  * - Repeated -m and -u opts overwrite one another -- again, not the point
  *   of this project and could be implemented with more time.
  * - Unlike find, this doesn't escape some chars (e.g., backslash and space),
  *   so those show up on the diff.
  * - Some of the output messages were changed a little to be more consistent
  *   and similar to the output of find.
  */
```

EXAMPLE OUTPUT

```
(base) [jon@archijon testfs]$ # generating some directory structure
(base) [jon@archijon testfs]$ # demonstrate symlinks, hard links (same inode #s)
(base) [jon@archijon testfs]$ whoami
jon
(base) [jon@archijon testfs]$ mkdir dir1 dir2 dir3
(base) [jon@archijon testfs]$ touch dir1/f1 dir1/f2 dir2/f3
(base) [jon@archijon testfs]$ ln -s dir1/f1 dir3/f6
(base) [jon@archijon testfs]$ ln dir1/f2 dir3/f7
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # show rls on some directories, invalid dir
(base) [jon@archijon testfs]$ find -ls
  5140990      4 drwxr-xr-x   5 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir1/f2
  6034874      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir1/f1
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir3/f7
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$ ./rls
  5140990      4 drwxr-xr-x   5 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir1/f2
  6034874      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir1/f1
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir3/f7
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$ ./rls .. | tail -n 5
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ../testfs/dir3/f6 -> dir1/f1
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ../testfs/rls
  1979747      0 lrwxrwxrwx   1 jon      jon             4 Sep 27 22:14 ../test2 -> test
  1978833     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:26 ../rls
  1979582     44 -rw-r--r--   1 jon      jon         41979 Sep 30 22:28 ../t1
(base) [jon@archijon testfs]$ ./rls dir3/../../testfs/dir2
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir3/../../testfs/dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 dir3/../../testfs/dir2/f3
```

```
(base) [jon@archijon testfs]$ ./rls mkldmd
rls: ERROR: opening directory "mkldmd": No such file or directory
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # demonstrate permissions
(base) [jon@archijon testfs]$ mkdir perms perms/forbidden perms/notraverse && cd perms
(base) [jon@archijon perms]$ touch 0777 0744 0000 1050 7000 4000 1000 7011 forbidden/test notraverse/test
(base) [jon@archijon perms]$ chmod 0777 0777
(base) [jon@archijon perms]$ chmod 0744 0744
(base) [jon@archijon perms]$ chmod 0000 0000
(base) [jon@archijon perms]$ chmod 1050 1050
(base) [jon@archijon perms]$ chmod 7000 7000
(base) [jon@archijon perms]$ chmod 4000 4000
(base) [jon@archijon perms]$ chmod 1000 1000
(base) [jon@archijon perms]$ chmod 7011 7011
(base) [jon@archijon perms]$ chmod 7666 forbidden
(base) [jon@archijon perms]$ chmod 7333 notraverse
(base) [jon@archijon perms]$ cd .. && ./rls perms
  6034878      4 drwxr-xr-x   4 jon      jon          4096 Oct  2 15:31 perms
  6034985      0 ---------T   1 jon      jon             0 Oct  2 15:31 perms/1000
  6034986      0 ---S--s--t   1 jon      jon             0 Oct  2 15:31 perms/7011
  6034883      0 ----------   1 jon      jon             0 Oct  2 15:31 perms/0000
  6034885      0 ---S--S--T   1 jon      jon             0 Oct  2 15:31 perms/7000
  6034879      4 drwSrwSrwT   2 jon      jon          4096 Oct  2 15:31 perms/forbidden
rls: stat "perms/forbidden": Permission denied
  6034881      0 -rwxrwxrwx   1 jon      jon             0 Oct  2 15:31 perms/0777
  6034880      4 d-ws-ws-wt   2 jon      jon          4096 Oct  2 15:31 perms/notraverse
rls: opening directory "perms/notraverse": Permission denied
  6034886      0 ---S------   1 jon      jon             0 Oct  2 15:31 perms/4000
  6034882      0 -rwxr--r--   1 jon      jon             0 Oct  2 15:31 perms/0744
  6034884      0 ----r-x--T   1 jon      jon             0 Oct  2 15:31 perms/1050
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # demonstrate -v
(base) [jon@archijon testfs]$ mkisofs -o d1iso.iso dir1
Setting input-charset to 'UTF-8' from locale.
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
174 extents written (0 MB)
(base) [jon@archijon testfs]$ mkdir mnt
(base) [jon@archijon testfs]$ sudo mount -o loop d1iso.iso mnt
```

```
mount: /home/jon/Documents/coursework/ece357/hw/programs/testfs/mnt: WARNING: device write-protected, mounted read-only.
(base) [jon@archijon testfs]$ ./rls
  5140990      4 drwxr-xr-x   7 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir1/f2
  6034874      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir1/f1
     1472      2 dr-xr-xr-x   1 root     root         2048 Oct  2 15:31 ./mnt
     1474      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f1
     1475      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f2
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir3/f7
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140992    348 -rw-r--r--   1 jon      jon        356352 Oct  2 15:31 ./d1iso.iso
  6034878      4 drwxr-xr-x   4 jon      jon          4096 Oct  2 15:31 ./perms
  6034985      0 ---------T   1 jon      jon             0 Oct  2 15:31 ./perms/1000
  6034986      0 ---S--s--t   1 jon      jon             0 Oct  2 15:31 ./perms/7011
  6034883      0 ----------   1 jon      jon             0 Oct  2 15:31 ./perms/0000
  6034885      0 ---S--S--T   1 jon      jon             0 Oct  2 15:31 ./perms/7000
  6034879      4 drwSrwSrwT   2 jon      jon          4096 Oct  2 15:31 ./perms/forbidden
rls: stat "./perms/forbidden": Permission denied
  6034881      0 -rwxrwxrwx   1 jon      jon             0 Oct  2 15:31 ./perms/0777
  6034880      4 d-ws-ws-wt   2 jon      jon          4096 Oct  2 15:31 ./perms/notraverse
rls: opening directory "./perms/notraverse": Permission denied
  6034886      0 ---S------   1 jon      jon             0 Oct  2 15:31 ./perms/4000
  6034882      0 -rwxr--r--   1 jon      jon             0 Oct  2 15:31 ./perms/0744
  6034884      0 ----r-x--T   1 jon      jon             0 Oct  2 15:31 ./perms/1050
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$ ./rls -v
  5140990      4 drwxr-xr-x   7 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir1/f2
  6034874      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir1/f1
     1472      2 dr-xr-xr-x   1 root     root         2048 Oct  2 15:31 ./mnt
rls: mount point "./mnt": Not crossing mount point
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034875      0 -rw-r--r--   2 jon      jon             0 Oct  2 15:31 ./dir3/f7
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140992    348 -rw-r--r--   1 jon      jon        356352 Oct  2 15:31 ./d1iso.iso
  6034878      4 drwxr-xr-x   4 jon      jon          4096 Oct  2 15:31 ./perms
  6034985      0 ---------T   1 jon      jon             0 Oct  2 15:31 ./perms/1000
  6034986      0 ---S--s--t   1 jon      jon             0 Oct  2 15:31 ./perms/7011
  6034883      0 ----------   1 jon      jon             0 Oct  2 15:31 ./perms/0000
  6034885      0 ---S--S--T   1 jon      jon             0 Oct  2 15:31 ./perms/7000
  6034879      4 drwSrwSrwT   2 jon      jon          4096 Oct  2 15:31 ./perms/forbidden
rls: stat "./perms/forbidden": Permission denied
  6034881      0 -rwxrwxrwx   1 jon      jon             0 Oct  2 15:31 ./perms/0777
  6034880      4 d-ws-ws-wt   2 jon      jon          4096 Oct  2 15:31 ./perms/notraverse
rls: opening directory "./perms/notraverse": Permission denied
  6034886      0 ---S------   1 jon      jon             0 Oct  2 15:31 ./perms/4000
  6034882      0 -rwxr--r--   1 jon      jon             0 Oct  2 15:31 ./perms/0744
  6034884      0 ----r-x--T   1 jon      jon             0 Oct  2 15:31 ./perms/1050
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # demonstrate -m
(base) [jon@archijon testfs]$ # I run this all as a script, so pretty much the created
(base) [jon@archijon testfs]$ # directory structure and processes should show up here
(base) [jon@archijon testfs]$ sleep 2
(base) [jon@archijon testfs]$ echo test > dir1/f1
(base) [jon@archijon testfs]$ sleep 2
(base) [jon@archijon testfs]$ echo test > dir1/f2
(base) [jon@archijon testfs]$ sleep 2
(base) [jon@archijon testfs]$ ./rls -m -3
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir1/f2
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir3/f7
rls: stat "./perms/forbidden": Permission denied
rls: opening directory "./perms/notraverse": Permission denied
(base) [jon@archijon testfs]$ ./rls -m -5
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir1/f2
  6034874      4 -rw-r--r--   1 jon      jon             5 Oct  2 15:31 ./dir1/f1
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir3/f7
rls: stat "./perms/forbidden": Permission denied
rls: opening directory "./perms/notraverse": Permission denied
(base) [jon@archijon testfs]$ ./rls -m 5
  5140990      4 drwxr-xr-x   7 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
     1472      2 dr-xr-xr-x   1 root     root         2048 Oct  2 15:31 ./mnt
     1474      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f1
     1475      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f2
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140992    348 -rw-r--r--   1 jon      jon        356352 Oct  2 15:31 ./d1iso.iso
  6034878      4 drwxr-xr-x   4 jon      jon          4096 Oct  2 15:31 ./perms
  6034985      0 ---------T   1 jon      jon             0 Oct  2 15:31 ./perms/1000
  6034986      0 ---S--s--t   1 jon      jon             0 Oct  2 15:31 ./perms/7011
  6034883      0 ----------   1 jon      jon             0 Oct  2 15:31 ./perms/0000
  6034885      0 ---S--S--T   1 jon      jon             0 Oct  2 15:31 ./perms/7000
  6034879      4 drwSrwSrwT   2 jon      jon          4096 Oct  2 15:31 ./perms/forbidden
rls: stat "./perms/forbidden": Permission denied
  6034881      0 -rwxrwxrwx   1 jon      jon             0 Oct  2 15:31 ./perms/0777
  6034880      4 d-ws-ws-wt   2 jon      jon          4096 Oct  2 15:31 ./perms/notraverse
rls: opening directory "./perms/notraverse": Permission denied
  6034886      0 ---S------   1 jon      jon             0 Oct  2 15:31 ./perms/4000
  6034882      0 -rwxr--r--   1 jon      jon             0 Oct  2 15:31 ./perms/0744
  6034884      0 ----r-x--T   1 jon      jon             0 Oct  2 15:31 ./perms/1050
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$ # show that invalid -m is ignored:
(base) [jon@archijon testfs]$ ./rls -m amdk
  5140990      4 drwxr-xr-x   7 jon      jon          4096 Oct  2 15:31 .
  6034871      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir2
  6034876      0 -rw-r--r--   1 jon      jon             0 Oct  2 15:31 ./dir2/f3
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir1
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir1/f2
  6034874      4 -rw-r--r--   1 jon      jon             5 Oct  2 15:31 ./dir1/f1
     1472      2 dr-xr-xr-x   1 root     root         2048 Oct  2 15:31 ./mnt
     1474      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f1
     1475      0 -r-xr-xr-x   1 root     root            0 Oct  2 15:31 ./mnt/f2
  6034873      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 ./dir3
  6034875      4 -rw-r--r--   2 jon      jon             5 Oct  2 15:31 ./dir3/f7
  6034877      0 lrwxrwxrwx   1 jon      jon             7 Oct  2 15:31 ./dir3/f6 -> dir1/f1
  5140992    348 -rw-r--r--   1 jon      jon        356352 Oct  2 15:31 ./d1iso.iso
  6034878      4 drwxr-xr-x   4 jon      jon          4096 Oct  2 15:31 ./perms
  6034985      0 ---------T   1 jon      jon             0 Oct  2 15:31 ./perms/1000
  6034986      0 ---S--s--t   1 jon      jon             0 Oct  2 15:31 ./perms/7011
  6034883      0 ----------   1 jon      jon             0 Oct  2 15:31 ./perms/0000
  6034885      0 ---S--S--T   1 jon      jon             0 Oct  2 15:31 ./perms/7000
  6034879      4 drwSrwSrwT   2 jon      jon          4096 Oct  2 15:31 ./perms/forbidden
rls: stat "./perms/forbidden": Permission denied
  6034881      0 -rwxrwxrwx   1 jon      jon             0 Oct  2 15:31 ./perms/0777
  6034880      4 d-ws-ws-wt   2 jon      jon          4096 Oct  2 15:31 ./perms/notraverse
rls: opening directory "./perms/notraverse": Permission denied
```

```
  6034886      0 ---S------   1 jon      jon             0 Oct  2 15:31 ./perms/4000
  6034882      0 -rwxr--r--   1 jon      jon             0 Oct  2 15:31 ./perms/0744
  6034884      0 ----r-x--T   1 jon      jon             0 Oct  2 15:31 ./perms/1050
  5140991     24 -rwxr-xr-x   1 jon      jon         22512 Oct  2 15:31 ./rls
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # demonstrate -u
(base) [jon@archijon testfs]$ chmod 0244 dir1/f1
(base) [jon@archijon testfs]$ sudo chown alice dir1/f1
(base) [jon@archijon testfs]$ chmod 0040 dir1/f2
(base) [jon@archijon testfs]$ sudo chown bob dir1/f2
(base) [jon@archijon testfs]$ ./rls dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034875      4 ----r-----   2 bob      jon             5 Oct  2 15:31 dir1/f2
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$ ./rls -u jon dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034875      4 ----r-----   2 bob      jon             5 Oct  2 15:31 dir1/f2
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$ ./rls -u 1000 dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034875      4 ----r-----   2 bob      jon             5 Oct  2 15:31 dir1/f2
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$ ./rls -u bob dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$ ./rls -u alice dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
(base) [jon@archijon testfs]$ ./rls -u root dir1
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034875      4 ----r-----   2 bob      jon             5 Oct  2 15:31 dir1/f2
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$ # show invalid -u
(base) [jon@archijon testfs]$ ./rls -u askdmsad dir1
rls: processing parameter -u "askdmsad": User not found
  6034870      4 drwxr-xr-x   2 jon      jon          4096 Oct  2 15:31 dir1
  6034875      4 ----r-----   2 bob      jon             5 Oct  2 15:31 dir1/f2
  6034874      4 --w-r--r--   1 alice    jon             5 Oct  2 15:31 dir1/f1
(base) [jon@archijon testfs]$
(base) [jon@archijon testfs]$ # show that its output is similar to find -ls (except whitespace)
(base) [jon@archijon testfs]$ # show block, char devices
(base) [jon@archijon testfs]$ ./rls /dev | tail -n 20
     3094      0 crw--w----   1 jon      tty           4,2 Oct  2 14:09 /dev/tty2
     3093      0 crw-------   1 jon      tty          4,1 Oct  2 15:22 /dev/tty1
     3092      0 crw-rw----   1 root     tty        7,129 Oct  2 13:21 /dev/vcsa1
     3091      0 crw-rw----   1 root     tty         7,65 Oct  2 13:21 /dev/vcsu1
     3090      0 crw-rw----   1 root     tty          7,1 Oct  2 13:21 /dev/vcs1
     3089      0 crw-rw----   1 root     tty        7,128 Oct  2 13:21 /dev/vcsa
     3088      0 crw-rw----   1 root     tty         7,64 Oct  2 13:21 /dev/vcsu
     3087      0 crw-rw----   1 root     tty          7,0 Oct  2 13:21 /dev/vcs
     3086      0 crw--w----   1 root     tty          4,0 Oct  2 13:21 /dev/tty0
     3085      0 crw-------   1 root     root         5,1 Oct  2 13:21 /dev/console
     3084      0 crw-rw-rw-   1 root     tty          5,0 Oct  2 15:27 /dev/tty
     3083      0 crw-r--r--   1 root     root        1,11 Oct  2 13:21 /dev/kmsg
     3082      0 crw-rw-rw-   1 root     root         1,9 Oct  2 13:21 /dev/urandom
     3081      0 crw-rw-rw-   1 root     root         1,8 Oct  2 13:21 /dev/random
     3080      0 crw-rw-rw-   1 root     root         1,7 Oct  2 13:21 /dev/full
     3079      0 crw-rw-rw-   1 root     root         1,5 Oct  2 13:21 /dev/zero
     3078      0 crw-r-----   1 root     kmem         1,4 Oct  2 13:21 /dev/port
     3077      0 crw-rw-rw-   1 root     root         1,3 Oct  2 13:21 /dev/null
     3076      0 crw-r-----   1 root     kmem         1,1 Oct  2 13:21 /dev/mem
     2074      0 crw-------   1 root     root       10,63 Oct  2 13:21 /dev/vga_arbiter
(base) [jon@archijon testfs]$ diff -w <(./rls /dev) <(find /dev -ls)
(base) [jon@archijon testfs]$ # (since diff showed nothing, is same except whitespace)
```