Advanced Programming in the UNIX Environment

Week 03, Segment 1: All about stat(2)

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

stat(2)

```
#include <sys/stat.h>

int stat(const char *path, struct stat *sb);
int lstat(const char *path, struct stat *sb);
int fstat(int fd, struct stat *sb);


#include <sys/stat.h>
#include <fcntl.h>

int fstatat(int fd, const char *path, struct stat *sb, int flag);
```

Returns: 0 if OK, -1 on error

All of these obtain information about the file pointed to by *path* (or, in the case of fstat(2), *fd*). If *path* is a symlink, lstat(2) returns information about the link itself.

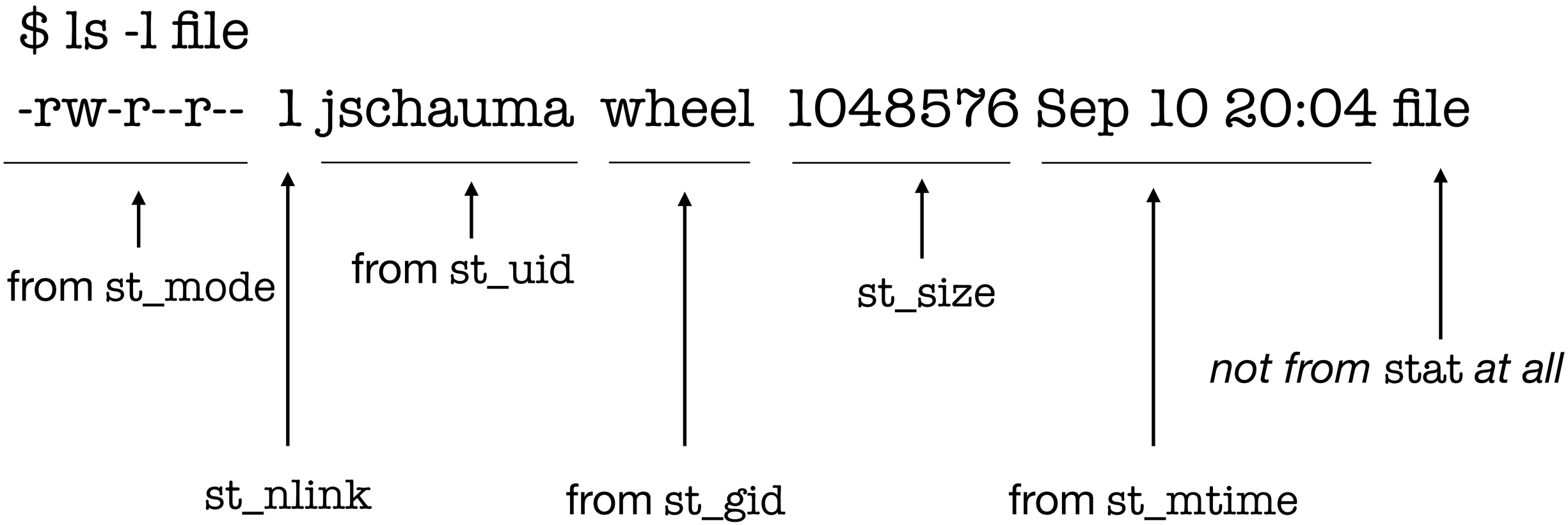
struct stat

```
struct stat  {
                dev_t    st_dev;      /* device number (filesystem) */
                ino_t    st_ino;      /* i-node number (serial number) */
                mod_t    st_mode;     /* file type & mode (permissions) */
                dev_t    st_rdev;     /* device number for special files */
                nlink_t  st_nlink;    /* number of links */
                uid_t    st_uid;      /* user ID of owner */
                gid_t    st_gid;      /* group ID of owner */
                off_t    st_size;     /* size in bytes, for regular files */
                time_t   st_atime;    /* time of last access */
                time_t   st_mtime;    /* time of last modification */
                time_t   st_ctime;    /* time of last file status change */
                long     st_blocks;   /* number of 512-byte blocks allocated */
                long     st_blksize;  /* optimal I/O block size */
}
```

Additional fields may be defined; check your system's manual page.

```
apue$ sudo newfs -b 4096 /dev/rwd1a
/dev/rwd1a: 50.0MB (102400 sectors) block size 4096, fragment size 512
        using 5 cylinder groups of 10.00MB, 2560 blks, 4832 inodes.
super-block backups (for fsck_ffs -b #) at:
32, 20512, 40992, 61472, 81952,
apue$ sudo mount /dev/wd1a /mnt
apue$ sudo chown jschauma /mnt
apue$ cd /mnt
apue$ df .
Filesystem   512-blocks       Used      Avail %Cap Mounted on
/dev/wd1a         96207          1      91396   0% /mnt
apue$ dd if=/dev/zero of=file bs=1024 count=1024
1024+0 records in
1024+0 records out
1048576 bytes transferred in 0.007 secs (149796571 bytes/sec)
apue$ ls -l file
-rw-r--r--  1 jschauma  wheel  1048576 Sep 10 20:04 file
apue$
```

stat(2)


$ ls -l file
-rw-r--r-- 1 jschauma wheel 1048576 Sep 10 20:04 file
from st_mode
st_nlink
from st_uid
from st_gid
st_size
from st_mtime
not from stat at all

stat(2)

The stat(1) utility displays information about the file pointed to by *file*.

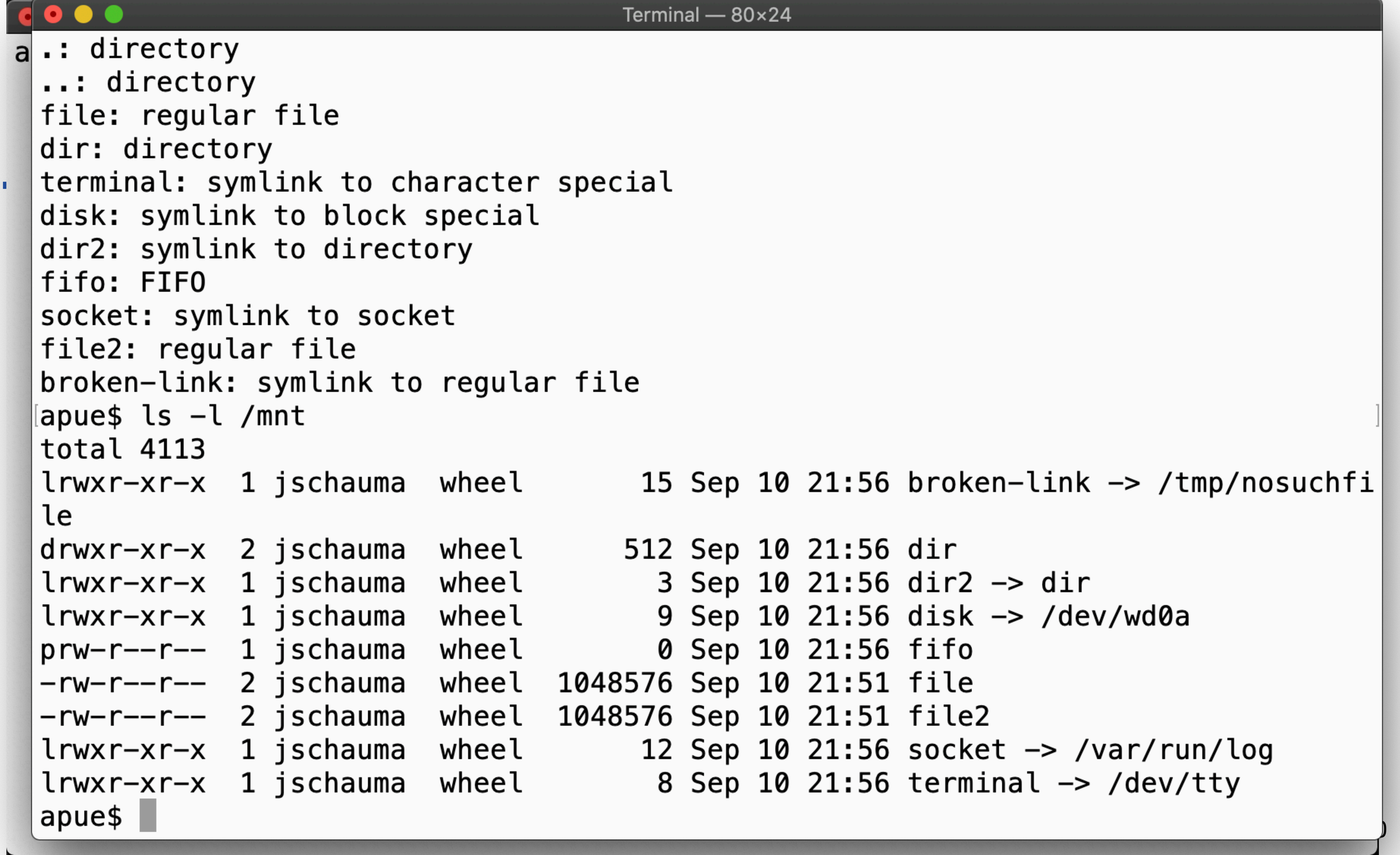
```
$ stat -r file
16 3 0100644 1 1000 0 -1 1048576 1599768274 1599768274 1599768274 0 4096 2056 0 file
```

The default format displays the st_dev, st_ino, st_mode, st_nlink, st_uid, st_gid, st_rdev, st_size, st_atime, st_mtime, st_ctime, st_birthtime, st_blksize, st_blocks, and st_flags fields, in that order.

struct stat: st_mode

The st_mode field of the struct stat encodes the *type* of file:

- **regular** – most common, interpretation of data is up to application
- **directory** – contains names of other files and pointer to information on those files
- **character special** – used for certain types of devices, e.g., terminal
- **block special** – used for disk devices (typically)
- **FIFO** – used for interprocess communication (sometimes called a "named pipe")
- **socket** – used for network communication and non-network communication (same host)
- **symbolic link** – points to another file


Terminal — 80×24
.: directory
..: directory
file: regular file
dir: directory
terminal: symlink to character special
disk: symlink to block special
dir2: symlink to directory
fifo: FIFO
socket: symlink to socket
file2: regular file
broken-link: symlink to regular file
apue$ ls -l /mnt
total 4113
lrwxr-xr-x 1 jschauma wheel   15 Sep 10 21:56 broken-link -> /tmp/nosuchfi
le
drwxr-xr-x 2 jschauma wheel   512 Sep 10 21:56 dir
lrwxr-xr-x 1 jschauma wheel   3 Sep 10 21:56 dir2 -> dir
lrwxr-xr-x 1 jschauma wheel   9 Sep 10 21:56 disk -> /dev/wd0a
prw-r--r-- 1 jschauma wheel   0 Sep 10 21:56 fifo
-rw-r--r-- 2 jschauma wheel 1048576 Sep 10 21:51 file
-rw-r--r-- 2 jschauma wheel 1048576 Sep 10 21:51 file2
lrwxr-xr-x 1 jschauma wheel   12 Sep 10 21:56 socket -> /var/run/log
lrwxr-xr-x 1 jschauma wheel   8 Sep 10 21:56 terminal -> /dev/tty
apue$

```
Terminal — 80×24
     -x      Multi-column output sorted across the page rather than down the
             page.
apue$ ls -ls file
20 -rw-r--r--  2 jschauma  wheel  10240 Sep 10 23:07 file
apue$ BLOCKSIZE=4096 ls -ls file
3 -rw-r--r--  2 jschauma  wheel  10240 Sep 10 23:07 file
apue$ df .
Filesystem   512-blocks       Used      Avail %Cap Mounted on
/dev/wd1a         96207         22      91375   0% /mnt
apue$ BLOCKSIZE=4096 df .
Filesystem  4096-blocks       Used      Avail %Cap Mounted on
/dev/wd1a         12025          2      11421   0% /mnt
apue$ ls -l
total 41
lrwxr-xr-x  1 jschauma  wheel     15 Sep 10 21:56 broken-link -> /tmp/nosuchfile
drwxr-xr-x  2 jschauma  wheel    512 Sep 10 21:56 dir
lrwxr-xr-x  1 jschauma  wheel      3 Sep 10 21:56 dir2 -> dir
lrwxr-xr-x  1 jschauma  wheel      9 Sep 10 21:56 disk -> /dev/wd0a
prw-r--r--  1 jschauma  wheel      0 Sep 10 21:56 fifo
-rw-r--r--  2 jschauma  wheel  10240 Sep 10 23:07 file
-rw-r--r--  2 jschauma  wheel  10240 Sep 10 23:07 file2
lrwxr-xr-x  1 jschauma  wheel     12 Sep 10 21:56 socket -> /var/run/log
lrwxr-xr-x  1 jschauma  wheel      8 Sep 10 21:56 terminal -> /dev/tty
apue$
```

struct stat

We've met our new best friend, the `struct stat`.

We've seen how `ls(1)` can display most of the information from the `struct stat`. `stat(1)` gets us the rest in a more flexible way.

We've revisited the `st_blocks` and `st_blksize` members from our discussion on I/O efficiency.

We improved our `simple-ls` clone to display the type of file via the `st_mode` for files and symlinks.

An yet, there's so much more to come...