pipe (I/O subsystem)

xv6 rev7

Carmi Merimovich

CS School, Tel Aviv Academic College

January 2,2017

pipe struct

```
struct pipe {
 struct spinlock lock;
 char data[PIPESIZE];
 uint nread; // number of bytes read
 uint nwrite; // number of bytes written
 int readopen; // read fd is still open
 int writeopen; // write fd is still open
};
```

- When reading, if the pipe is empty then we should wait for the event "there are bytes in the pipe".
- When writing, if the pipe is full then we should wait for the event "there is space in the pipe".

Hence:

- On successfull read we should declare "there is space in the pipe".
- On successfull write we should declare "there are bytes in the pipe".

pipealloc

```
6020  int pipealloc(struct file **f0, struct file **f1) {
      struct pipe *p = 0 ;
      *f0 = *f1 = 0;
      if ((*f0=filealloc())==0||(*f1=filealloc())==0) goto bad
      if ((p = (struct pipe*)kalloc()) == 0) goto bad;
      p->readopen = 1;
      p->writeopen = 1;
      p->nwrite = 0;
      p->nread = 0;
      initlock(&p->lock, "pipe");
      (*f0)->type = FD_PIPE;
      (*f0)->readable = 1;
      (*f0)->writable = 0;
      (*f0)->pipe = p;
      (*f1)->type = FD_PIPE;
      (*f1)->readable = 0;
      (*f1)->writable = 1;
      (*f1)->pipe = p;
      return 0;
```