

CS241 #23 Files Pipes Seeks II

1. How does the C library wrap a filedescriptor?

```
01 // How the C library wraps a filedescriptor
02 typedef struct _FILE { /* Simplified!*/
03     int fd;
04     void* buffer; // reduce # of write() calls
05     size_t buf_capacity, buf_size;
06     int mode; // _IONBF / _IOLBF / _IOFBUF
07 } FILE;
```

fprintf(FILE*, format,...)

uses write (buffering reduces number of writes => performance)

fseek(FILE*, offset, whence) SEEK_CUR | SEEK_SET | SEEK_END

uses lseek(int fd, off_t offset, int whence);

long pos=ftell(FILE*) uses return lseek(fd, 0, SEEK_CUR)

2. Challenge: Implement C library function **rewind(FILE*)**

Hint: You will need fflush and fseek

```
01 // Use the struct above to extract the filedes
02 void rewind(FILE* f) {
03
04
05 }
```

3. Reading & Writing binary data

fread (void * ptr, size_t size, size_t nitems, FILE * stream);

fwrite (void * ptr, size_t size, size_t nitems, FILE * stream);

```
01 int num_pts;
02 typedef struct { float x,y,z } p_t;
03 p_t* points;
04
05 void load_point_cloud() {
06     FILE* f = fopen("points.dat","r");
07     fread( &num_pts , _____ , 1, f);
08     points = calloc( sizeof p_t, num_pts);
09     ?
10
```

Error handling/What could go wrong? Why #include<stdint.h> and using uint32_t be better?

4. Challenge: Read in the first half of a file as C string

Hints: fopen, fseek, ftell, fread, malloc, fclose may be useful

```
01 char* half(char*filename) {
02
03
04
05
06
07
08
```

5. Implement **fflush**

Hint: Use & reset the FILE's output buffer, write will be useful

```
01 void fflush(FILE*f) {
02
03
04
05
```

6. Challenge: Parent process will copy the contents (4KB at a time) of a file 'input.txt' into stdin of the child process that exec's a bash shell
Hint: open, pipe, read returns 0, write, fork, dup2(int current_fd, int new_fd); Assume read and write always complete.

7. Quiz 3 review