209 - Tutorial Week 6

Files and directories in C

Opening files

FILE *fopen(const char *filename, const char *mode);

- Filename: identifies file to open
- Mode:
  - “r” for reading
  - “w” for writing
  - “a” for appending

Closing files

```
int fclose(FILE *stream);
```

File Output

int fprintf(FILE *stream, const char *format, …);

- Like printf, except for the first argument that identifies the file to write to.
- Example:

fprintf(stdout, “Writing to standard output\n”);

File Input

char *fgets(char *s, int size, FILE *stream);

- Reads one line from a file.
- Stops at newline or EOF.

Other Input/Output library calls

- int fputc(int c, FILE *stream);
- int fputs(char *s, FILE *stream);
- size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
- size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

Directories

- Recall that directory entries are really stored in a file.
- Opening:

  DIR *opendir(const char *filename);
- Example:

  DIR *d;

  d = opendir(“myfile”);

Reading entries

struct dirent *readdir(DIR *dir);

- struct dirent{

  long d_ino;

  off_t d_off;

  unsigned short d_reclen;

  char d_name[NAME_MAX + 1];

  }

Output directory content

```
DIR *dp;
struct dirent *d;
/* open the current working directory */
if((dp = opendir(".")) == NULL){
    perror("opendir");
    exit(1);
}
while((d = readdir(dp)) != NULL) {
    printf("%s\n", d->d_name);
}
closedir(dp);
```

printdirs.c

```
#include <stdio.h>
#include <dirent.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>

int main(int argc, char **argv){
    int i;
    struct stat sbuf;
    char fullname[NAME_MAX];
    if(argc < 2) {
            fprintf(stderr, "Usage: printdirs [dir ...]\n");
            exit(1);
    }
```

printdirs.c (2)

```
for(i = 1; i < argc; i++) {
    DIR *dp = opendir(argv[i]);
    struct dirent *entry;
    if (dp == NULL) {
        perror(argv[i]);
        continue;
    }
    while((entry = readdir(dp)) != NULL) {
        strncpy(fullname, argv[i], NAME_MAX);
        strcat(fullname, "/");
        strcat(fullname, entry->d_name);
        if(stat(fullname, &sbuf) == -1) {
                perror(entry->d_name);
                continue;
        }
        if(S_ISDIR(sbuf.st_mode)) {
                printf("%s\n", fullname);
        }
    }
    closedir(dp);
}
return 0;}
```