

Lecture 30 (4-10-26)

System Calls, File System, I/O (System Calls)

Motivating Questions:

1. What is a **system call**?
 2. How do we lookup a file's **inode** information?
 3. How do we walk a **directory**?
 4. How do we copy a file's **data**?
-

list.c

```
/* list.c */

#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <dirent.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

/* Node Structure */

typedef struct Node Node;
struct Node {
    char *path;
    size_t size;
    Node *next;
};

// TODO

Node * node_create(const char *path, size_t size) {
```

```

// TODO
Node *n = calloc(1, sizeof(Node));
if (n){
    n->path = (char *)path;
    n->size = size;
}
return n;
}

void    node_delete(Node *n) {
    // TODO
    if (!n){return;}
    node_delete(n->next);
    free(n->path);
    free(n);
}

/* List Structure */

// TODO
typedef struct{
    Node *head;
}List;

void    list_add(List *l, const char *path, size_t size) {
    // TODO
    Node *n = node_create(path, size);
    // Case 1: Empty List
    if (!l->head){
        l->head = n;
    } else { // Case 2: Non-Empty List
        Node *tail = l->head;

        while (tail->next){
            tail = tail->next;
        }

        tail->next = n;
    }
}

/* Functions */

void    add_files(const char *root, List *files) {
    // TODO

```

```

DIR *d = opendir(root);
if (!d){
    fprintf(stderr, "opendir(%s): %s\n", path, strerror(errno));
    return;
}

for (struct dirent *e = readdir(d); e; e = readdir(d)){
    if (strcmp(e->d_name, "." == 0 || strcmp(e->d_name, ".." == 0){
        continue;
    }

    char path[BUFSIZ];
    sprintf(path, '%s/%s', root, e->d_name); //combines name with the
root

    struct stat *s;
    if (stat(path, &s) != 0){
        fprintf(stderr, "stat(%s): %s\n", e->d_name, strerror(errno));
        continue;
    }

    if (s.st_mode &S_IFMT != S_IFREG){
        continue;
    }

    if (access(path, XOK) != 0){
        continue;
    }

    list_add(files, strdup(path), s.st_size);

    printf("[%8lu] %s\n", s.st_size, e->d_name);
}

closedir(d);
}

void print_files(List *files) {
    // TODO
    for (Node *curr = files->head; curr; curr = curr->next){
        printf("[%lu] $s\n", curr->size, curr->path);
    }
}

/* Main Execution */

```

```

int main(int argc, char *argv[]) {
    // TODO: Determine directory
    char root[BUFSIZ];

    if (argc >= 2){
        strcpy(root ,argv[1]);
    } else {
        getcwd(root, BUFSIZ); //system call!!!!
    }
    printf("root is %s\n", root);

    // TODO: Add files and print them
    List l = {NULL};
    add_files = (root, &files);
    print_files(&files);

    // TODO: Release resources
    node_delete(l->head)

    return EXIT_SUCCESS;
}

/* vim: set sts=4 sw=4 ts=8 expandtab ft=c: */

```

I/O: Overview

- All **input** and **output** is performed on **files**:
1. **Open**
 - Create handle to stream of data
 2. **Close**
 - Destroy handle to stream of data
 3. **Read**
 - Retrieve chunk from stream of data
 4. **Write**
 - Append chunk to stream of data
 5. **Seek**
 - Move within stream of data