HW6

lrand48_r

> lrand48_r
> 从一个buffer里读东西，输出**31**bits 随机字符到result里
> 因此需要srand48_r吃一段数据作为seed，先对一个buffer生成随机字符
> 再拿这个buffer作为lrand48_r的输入
> 注意目标需要64bits 随机字符，所以要位运算3次

```
lrand48_r(struct drand48_data *buffer, long int* result):

strict drand48_data buffer;
srand48_r(12345, &buffer);

long int rand1, rand2, rand3;
lrand48_r(&buffer, &rand1);
lrand48_r(&buffer, &rand2);
lrand48_r(&buffer, &rand3);

unsigned long long rand64 = (rand1 << 33) | (rand2 << 2) | (rand2 && 0x3);
```

mrand48:

> mrand48
> 直接输出32bits 随机字符
> 因此位运算2次

```
long int rand1 = mrand48();
long int rand2 = mrand48();

long long rand64 = (rand1 << 32) | rand2;
```

getopt

```
#include "options.h"

long long process_arguments(int argc, char** argv, option* options) {
    int opt;
    while ((opt = getopt(argc, argv, "i:o:")) != -1) {
        switch (opt) {
            case 'i':
                options->input = 1;
                if (strcmp(optarg, "") == 0) {
```

```
                    fprintf(stderr, "Expected one argument for the input
source.\n");
                    return -1;
                } else {
                    options->input_source = optarg;
                }
                break;

            case 'o':
                options->output = 1;
                if (strcmp(optarg, "") == 0) {
                    fprintf(stderr, "Expected one argument for the output
source.\n");
                    return -1;
                } else {
                    options->output_source = optarg;
                }
                break;

            default:
                break;
        }
    }

    if (optind >= argc) {
        fprintf(stderr, "Expected one non-option argument for the number
of bytes.\n");
        return -1;
    }

    errno = 0;
    char *endptr;
    long long nbytes = strtoll(argv[optind], &endptr, 10);

    if (errno != 0 || *endptr != '\0' || nbytes < 0) {
        fprintf (stderr, "%s: usage: %s NBYTES\n", argv[0], argv[0]);
        return -1;
    }

    return nbytes;
}
```

> optind 直接指向non-optional argument after the while(opt) thus, it is refering to a `nbytes` in this assignment

Checklist

Here's the final study list shared today in Dis 1C (for some reason discord isn't allowing me to upload the file):

Final

When: Mon 8am Where: Ackerman

HW1: Emacs and Shell scripting

- Keyboard shortcuts
- Shell scripting
    - Syntax
        - Shebang ("#!/bin/bash")
        - if, else
        - Variable assignment
        - Arguments, special variables
        - Redirection
    - Programs/commands
        - ps
        - tr
        - diff, comm
        - awk
        - sed
        - find
        - cat
        - echo
        - touch
        - grep
    - Regex
        - POSIX Extended
        - grep -E
    - File systems stuff
        - Symlinks vs hardlinks
        - inodes

HW2: Python and elisp

- Opening files
- Data structures (list, dict, tuples, sets)
- Coding (python and elisp)

HW3: React and testing

- Components and rendering, states, sync and async
- npm (package management), package.json, what to include in git
- Network protocols and caching
- Testing

HW4: Git Usage

- Commands:
    - init, clone, pull, push
    - branch, add, commit, merge, reset, checkout

HW5: Git Internals, Topological Sort

- trees, what is in each node
  - child -> parent
- Ordering of dependencies

HW6: C and makefiles

- valgrind: detect memory issues
- gdb: breakpoints, watchpoints, backtrace, checkpoints, reverse continue
- gcc flags
  - -g: add debugging info
  - -O: optimization (-O1, -O2, -O3, -Os)
  - -fsanitize:
  - -flto:
- Makefiles:
  - syntax (target, dependencies, rules)
  - special variables ($@, $^)
  - When does make run a command: dependencies newer than target