CSC 161

Fall 2023

Function Pointers & File IO

Announcements

- Exam #2: graded this weekend, back on Monday
  - Recall: you can use an extra credit token to *revise* one question on the midterm!
- Schedule cadence will be the same!
  - Next homework #5 out on Monday, due *next Wednesday* (before break)
  - No lab turn-in/quiz today—starts up again next week
- Reminder: no *in-person* class on 11/22 (before break)
  - (But there will be a reading!)
- Office hours (starting next week, Prof. Eikmeier's office, 3809)
  - Mondays, Thursdays, and Fridays: 10:30–11:30
  - Wednesdays: 2–2:30

Function Pointers

Function Pointers: Overview

- Why might we want to pass functions to other functions?

```
(lambda (x) (+ x 1))
```

- Write down local declarations of variables of function pointer type that can point to each of each of the following functions:
  - `int increment(int x)`
    - `int (*fp1) (int) = &increment;`
  - `void modify (int *out, char *str)`
    - `void (*fp2) (int*, char*) = &modify`

- What is the syntax for invoking a function through a function pointer?
  - `(*fp1)(10)`

- *Is there anything we* can't *do with a function pointer in C that we can do with a lambda in Scheme?*
  - *"Capture" in-scope, non-local variables!*

Example: Comparators

```c
void selection_sort(int *arr, const size_t len) {
  for (int i = 0; i < len; i++) {
    int min_index = i;
    for (int j = i + 1; j < len; j++) {
      if (arr[j] < arr[min_index]) {
        min_index = j;
      }
    }
    int tmp = arr[i];
    arr[i] = arr[min_index];
    arr[min_index] = tmp;
  }
}

// ...
int arr[] = { 7, 8, 2, 4, 1, 8, 9, 2, 0, 3};
selection_sort(arr, 10);
```

```c
typedef int (*comparator)(int, int);

void selection_sort(comparator comp, int *arr, const size_t len) {
  for (int i = 0; i < len; i++) {
    int min_index = i;
    for (int j = i + 1; j < len; j++) {
      if (comp(arr[j], arr[min_index]) < 0) {
        min_index = j;
      }
    }
    int tmp = arr[i];
    arr[i] = arr[min_index];
    arr[min_index] = tmp;
  }
}

int gt_comp(int x, int y) { return x - y; }

// ...
int arr[] = { 7, 8, 2, 4, 1, 8, 9, 2, 0, 3};
selection_sort(&gt_comp, arr, 10);
```

File I/O

File Input/Output: Overview

- Which header contains functions relevant for manipulating files?
  - `stdio.h` (*i.e.*, "stud"-IO-dot-h! 😀)
- What is the type of a "file" in C?
  - `FILE*`
- How do I open a file in C?
  - `FILE * fopen ( const char * filename, const char * mode );`
- What are the different functions I can use to read from/write to a file? When would I use function one versus another?
  - Reading one character at a time:
    - `int fgetc ( FILE * stream );`
  - Reading a (up to fixed-size) string:
    - `char * fgets ( char * str, int num, FILE * stream );`
  - Writing a string
    - `int fprintf ( FILE * stream, const char * format, ... );`
  - Binary reading/writing:
    - `size_t fread ( void * ptr, size_t size, size_t count, FILE * stream );`
    - `size_t fwrite ( const void * ptr, size_t size, size_t count, FILE * stream );`
- How do I "clean up" after I'm done reading/writing to a file?
  - `FILE * fopen ( const char * filename, const char * mode );`

Lab: File I/O