

Program 4b: getopt_long()
CSCI 4547 / 6647 Systems Programming
Fall 2017

1 Goals

1. To finish program 4 by including and processing long options.

2 Getopt_long()

3 Instructions

Write a program to parse the command line for DiskSweeper. Due Sept. 22 (Friday).

Add to your Params class. Members of the class should include:

- A bool variable for the debug option and an int for the optional level number.
- A bool variable for the verbose option.

Define a controller class named Sweeper. Members of the class should include:

- An Params object.
- A constructor that accepts argc and argv from main. This constructor must parse the command line and initialize the Params object.
- Anything else you need. You will be adding members to this class each week.
- A run() function: Declare an instance of Params and pass argc and argv to its constructor. When construction is finished, call Params::print() to display the params.

In your main function:

- Call banner() from tools.cpp.
- Create a Sweeper object.
- Call its run function.
- Nothing else.

4 Using getopt_long() to Decode Command-line Arguments

You still need nearly all of what you wrote for Program 4a. Here are the changes. For more guidance, refer to options2.c

- The call has two more parameters:

```
int = getopt_long( int argc, char* argv[], const char* opts,
                  struct option longOpts[], int*);
```
- You must pick up and process any additional options and arguments.

- You need to define a long options table. Example:

```
struct option longOpts[] = {
    { "verbose",    no_argument,       NULL, 'b' },
    { "output",     required_argument, NULL, 'o' },
    { "recursive",  no_argument,       NULL, 'R' },
    { "debug",      optional_argument, NULL,  0  },
    { NULL,         0,                 NULL,  0  }
};
```

- Add this statement to your switch to process long options (such as debug) that do not have corresponding short options.

```
case 0: { // Convert code for long switch to full name.
    outputLong(longOpts[code].name, optarg);
    break;
};
```