Lecture #7: Debugging

Reading: *none*

Handouts: The Dynamic Debugger gdb; Guide to Faster, Less Frustrating Debugging

Homework: ***study for the midterm!***

---

1. Greetings and felicitations
2. Debugging
   - programs have bugs; find and fix them
   - static debugging: insert debugging code into source, recompile and run
   - dynamic debugging: look at the program as it runs, observing (and maybe changing) variables, etc.
3. Static debugging
   - using printf to print variable values; mention %p (prints pointer value, usually as a hex integer)
   - using printf to print where you are (ie, on function entry printf("in function\n");
   - #ifdef DEBUG … #endif around the printfs so you can leave them in the source if you need them again
   - assert(x) macro: assert(0 <= i && i <= n) causes program to exit with error message if (0 <= I && I <= n) is false; must include <assert.h>. To delete, say #define NDEBUG and they will not be in the compiled code.
4. Dynamic debugging
   - debugging tool instruments executable program so it can be stopped, examined, altered, and continued interactively
   - go through the handout
   - mention the "where" command which shows you the program stack