Large Programs

Lecture 10

1

Large Programs

- When programs are long
  - Need to split the file into several files
  - Need to compile them together
  - Example
    - main.c, list.c, file.c
    - cc -o proj main.c list.c file.c

2

Large Programs

- Sharing glabal variables
  - Define types in a .h file
  - Declare global variables (do not initialize!)
    - extern in a .h file
  - Declare and initialize global variables
    - in one of the .c files (e.g., the main one)
  - Include the .h file in all the .c files which use any of the global variables

    #include "proj.h"

3

Large Programs

- Example
  - In link.h, declare (do not initialize!) global variables

    extern NODE *head;
  - In main.c, include link.h, and declare and initialize global variables

    #include "link.h"

    NODE *head = (NODE *)NULL;
  - In all the other .c files, just include link.h

    #include "link.h"

4

Large Programs

- Use makefile to automate the compilation process
- Create a "makefile" file with instructions
- Command make will parse the makefile file and execute the appropriate command

  <prompt> make proj1

5

makefile

```
proj1: main1.c list1.c io1.c
       gcc -o proj1 main1.c list1.c io1.c
       ./proj1 filename

proj2: main2.c list2.c thr2.c
       gcc -o proj2 main2.c list2.c thr2.c -lpthread
       ./proj2 filename1 filename2
```

6

Development Tools

- Unix/Linux
  - Environment and commands
  - Tools – learn to use these!!
    - vi
    - grep
    - gdb
    - makefile
    - shell script

7

Development Tools

- Libraries
  - Math
  - String
  - Memory
  - Threads

8