Linux Kernel
History

• Unix – Late 60's Bell Labs
• GNU – Richard Stallman (mid 80's)
• Minix – Andrew Tanenbaum '86
• Linux – Linus Torvalds '92
• Interesting link:
A Member of the Unix Family...

Today

- Linus still maintains the code, works for non-profit Linux Foundation
- A note on development model...
- Numbering system... 2.2.6 etc.
- Current SLOC...

*IBM Link: Anatomy of the Linux Kernel*
The Source Tree

• Let's look back at IBM link to see what's in the kernel.

• Good description of source tree:

  http://www.linuxchix.org/content/courses/kernel_hacking/lesson6

• Link to Cross Reference:
  - http://lxr.linux.no/linux+v2.6.30.10/
Build Process

• Setting up .config file has two effects:
  – #define
  – Determines what source files are compiled

• Mysterious asm/ directory

• Let's look at a header file...
  – (No one to one correspondence between source files and header files.)
Reading/Writing Kernel Code

• These guys don't seem to be using malloc very much. Where is printf? What's going on here?
• Some useful stuff is in include/linux/string.h
• Elsewhere:
  – kmalloc
  – printf
Linked Lists

- include/linux/list.h
- Incidentally... Why so many macros?
C Miscellany

- **static:**
  - In a function: means that memory is persistent
  - Outside of a function, limits scope to the current file.

- **extern:**
  - Declares, but does not define the variable.
    - No memory is set aside, the variable will be defined somewhere else.
  - **extern functions:**
    extern is significant only with data declarations. In function declarations, it can be used as a stylistic hint to indicate that the function's definition is probably in another source file, but there is no formal difference between
    ```c
    extern int f();
    ```
    and
    ```c
    int f();
    ```