2. Survival Kit

Helper Tools for System Programming

Processes: `ps/pstree`

- All processes: `ps -ef` or `ps -efww` (full command line)
- Easier to use `pstree` to inspect child threads

System Call Trace: `strace`

- Traces the sequence of system calls (with their arguments and return status)
- Verbose, but fast and useful to debug concurrent programs

Open files: `lsof/fuser`

- List open files
- List processes using a specific file

Other Helper Tools

Debugger: `gdb`

- Read the technical appendix (*l'annexe*), page 93
- Always compile with `-g`

Project manager: `make`

- A `Makefile` is provided with the labs
- You are encouraged to learn how to write one and extend it if needed

Shell and Internationalization Tricks

- Learn the basic functions of your *shell*
  Which shell am I using? `$ echo $SHELL`
- Hint: switch to another language (to English a priori)
  With `tcsh`: `$ setenv LANG C`   or   `$ setenv LANG fr_FR`
  With `bash`: `$ export LANG=C`   or   `$ export LANG=fr_FR`
  List internationalizations: `$ locale -a | more`

C Language and Standard Library

This Is Not a Programming Course

You may like C or not...
You may like English or not...
There is not choice but to learn and practice it!

Getting Help

- Definitive reference: C language book by B. Kernighan and D. Ritchie

- Use printed and online C tutorial (see course web page)

Help Yourself

- Read `man` pages
  - `man 1 command` (UNIX command)
  - `man 2 system_call` (primitive system calls)
  - `man 3 library_call` (e.g., C library, system call front-end functions)
  - Warning: multiple entries with the same name may appear in different sections of the `man` pages
    $\rightarrow$ avoid running `man name` without section number if you are not sure
  - The `SEE ALSO` section at the bottom of most `man` pages important way to navigate through this valuable source of precise/reference information

- Read the technical appendix (*l'annexe*, simpler, more pedagogical)