Programming Reference

Textbooks (https://canvas.wisc.edu/courses/330348/pages/textbooks?wrap=1) See link for details on each textbook

- **Required Reading (Topics and Readings 1 (https://canvas.wisc.edu/courses/330348/pages/topics-and-readings-1?wrap=1) Topics and Readings 2 (https://canvas.wisc.edu/courses/330348/pages/topics-and-readings-2?wrap=1))**
  - C Programming Language, 2nd Edition
  - Computer Systems: A Programmer's Perspective, 2nd Edition

---

Linux Command-line Tools

Operating System commands you should know **linux_reference.pdf (https://canvas.wisc.edu/courses/330348/files/30411901?wrap=1) (https://canvas.wisc.edu/courses/330348/files/30411901/download?download_frd=1)**

- `man` -- user manual that is available from the command line with information about current version. **man pages help (https://web.stanford.edu/class/archive/cs/cs107/cs107.1218/resources/man)**
- `vim, emacs, pico` -- text-only editors that all CS students should know and be able to use
  - **vim Settings (https://canvas.wisc.edu/courses/330348/pages/vim-settings)**
- `gcc` -- GNU C Compiler (used to build executables from C source code)
- `gdb` -- use to step through executable
  - **GDB_Cheat_Sheet.pdf (https://canvas.wisc.edu/courses/330348/files/30412116?wrap=1) (https://canvas.wisc.edu/courses/330348/files/30412116/download?download_frd=1)**
  - **Video examples (https://canvas.wisc.edu/courses/330348/pages/gdb-tutorial-youtube-video)**
- `valgrind` **-- used to check for memory leaks**
  1. **p3 example:** `valgrind --leak-check=yes ./test_align1`
  2. **https://valgrind.org/docs/manual/quick-start.html (https://valgrind.org/docs/manual/quick-start.html)**
- `objdump` -- used on p5 executables to view the object code disassemble
- `kill` -- used in p6 to send signals to processes from the command line

---

Common C Libraries and Functions

- `#include stdio.h -- printf, scanf, fprintf, fscanf, fopen, fclose, fgets`
- `#include stdlib.h -- atoi, strtol, strtok, malloc, calloc, realloc, free`
- `#include string.h -- strlen, strcpy, strcat`

Caution: the above is not exhaustive list

GNU Formatted Output in C (https://www.gnu.org/software/libc/manual/html_node/Formatted-Output.html) -- additional info if you want to print more than characters, integer, addresses, strings, etc.

Programming Guides

- Programming Style Guide (https://canvas.wisc.edu/courses/330348/pages/programming-style-guide) **- must have a consistent and easy to read source code**
- Programming Commenting Guide (https://canvas.wisc.edu/courses/330348/pages/program-commenting-guide) - source code must have File Header, Function Headers, and comments that describe the high-level algorithm choices

All Program work must be completed on CSL Linux workstations -- Students may connect remotely from personal computers.

- Windows/Mac/Linux Users: Use your terminal app and `ssh` to connect via Secure SHell protocol
  - Launch terminal app on personal computer
  - `ssh cslogin@best-linux.cs.wisc.edu`
- MobaXterm (https://mobaxterm.mobatek.net/) terminal application is a free download and is not required but it lets users save session configurations

---

Additional Resources:

Still more programming reference manuals & development setup guides for those going on into computer architecture and systems.
These document provide much more detail than CS354 requires.

- **C programming language** *(only check these out if you want to know all the details about C from the source)*

  - GNU C reference manual [**pdf** (https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.pdf) , **source** (https://www.gnu.org/software/gnu-c-manual/)]: *describes the programming constructs defined by the C99 spec and implemented by the GCC (GNU C Compiler)*.
  - ISO/IEC C99 standard document: [**pdf** (https://open-std.org/JTC1/SC22/WG14/www/docs/n1256.pdf) , **source** (https://open-std.org/JTC1/SC22/WG14/www/projects#9899)]: *actual C99 specification, this includes specific details of language features and design choices.*
- **C standard library & C operating system libraries**
  - GNU C library (*glibc*) manual [**pdf** (https://www.gnu.org/software/libc/manual/pdf/libc.pdf) , **source** (https://www.gnu.org/software/libc/)]: *Built-in C standard library functions (ISO C standard) and Core libraries for GNU/Linux-based operating systems (including standard interfaces such as POSIX)*.
  - C library online documentation [**devdocs.io/c** (https://devdocs.io/c/)]: *includes multiple C standards.*
- **Manual Pages online** (*man* command) [**kernel.org/doc/man-pages** (https://www.kernel.org/doc/man-pages/) , **source** (https://man7.org/linux/man-pages/)]
  CAUTION: there is no guarantee this online man information is correct for your or the CSL machine you are working from. Use man from the command line to have the actual man page information for the Operating System and architecture you are working on.
- **Intel Architecture, 32-bit**
  - Intel 64 and IA-32 Architectures Software Developer Manuals [**pdf** (https://cdrdv2.intel.com/v1/dl/getContent/671200) , **source** (https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html)]: *Latest software developer's manual (complete volumes combined with 64-bit description). In the source PDF describing instruciton set can be downloaded separately.*
  - IA-32 Intel Architecture Software Developer's Manual [**vol. 1** (http://courses.cs.washington.edu/courses/cse548/05wi/files/IA32-Volume1.pdf) , **vol. 2** (http://courses.cs.washington.edu/courses/cse548/05wi/files/IA32-Volume2.pdf) , **vol. 3** (http://courses.cs.washington.edu/courses/cse548/05wi/files/IA32-Volume3.pdf) , **source** (https://www.semanticscholar.org/paper/IA-32-Intel-Architecture-Software-Developers-Manual-Corportation/4f780488c8d1790bb4e8ad476a4b874403c1c336)]: *Manual specific for 32-bit architecture.*