Unix tools review

CSE451 ’16 Spring

Git

- Distributed version control system
  - See https://git-scm.com/book/en/v2
  - previous tutorial: http://courses.cs.washington.edu/courses/cse451/14au/tutorials/tutorial_git.html
  - If you're interested in internals:
    - https://git-scm.com/book/en/v1/Git-Internals
- Some useful features
  - log, diff, stash, branch
  - See https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Git (note)

- **Push** commits once you have something to share
- Merge upstream branches when TAs notify changes from the original distribution
  - Check remote with git remote -v

```
origin git@gitlab.cs.washington.edu:syhan/16sp_cse451_os161.git (fetch)
origin git@gitlab.cs.washington.edu:syhan/16sp_cse451_os161.git (push)
upstream git@gitlab.cs.washington.edu:syslab/16sp_cse451_os161.git (fetch)
upstream git@gitlab.cs.washington.edu:syslab/16sp_cse451_os161.git (push)
```

  - git pull upstream master

Browsing Source Codes

- less
  - Terminal pager
  - Usually used with pipe (e.g., ls | less)
  - Navigate with space, b, j, k
  - Search with /, ?
- grep
  - -i case insensitive search
  - -r recursive
  - -v invert
  - Or git grep

Browsing Source Codes

- tree
  - Dump the directory hierarchy to stdout
  - tree | less
- find
  - File search tool
  - find <dir> -name ‘pattern’

Browsing Source Codes

- ctags (or etags)
  - Index symbols in the source code
  - In kern/compile/ASST?, run bmake ctags (or bmake etags)
- vim + ctags
  - set tags=./tags;
  - ctrl-] jumps to tag under cursor
  - :tag <tag> jumps to specific tag
  - ctrl-T jumps back
- emacs + etags
  - M-. <RET> jumps to tag under cursor
  - M-. <tag> jumps to specific tag
  - M-* jumps back

Debugging

- mips-harvard-os161-gdb
  - GDB for OS161
  - gdbinit
    - wget http://courses.cs.washington.edu/courses/cse451/16sp/gdbinit && mv gdbinit ~/.gdbinit
  - change the directory according to your compilation dir
- Running GDB
  - sys161 -w kernel (waiting gdb)
  - mips-harvard-os161-gdb
    - target remote unix:.sockets/gdb
  - Check out -tui option

Debugging

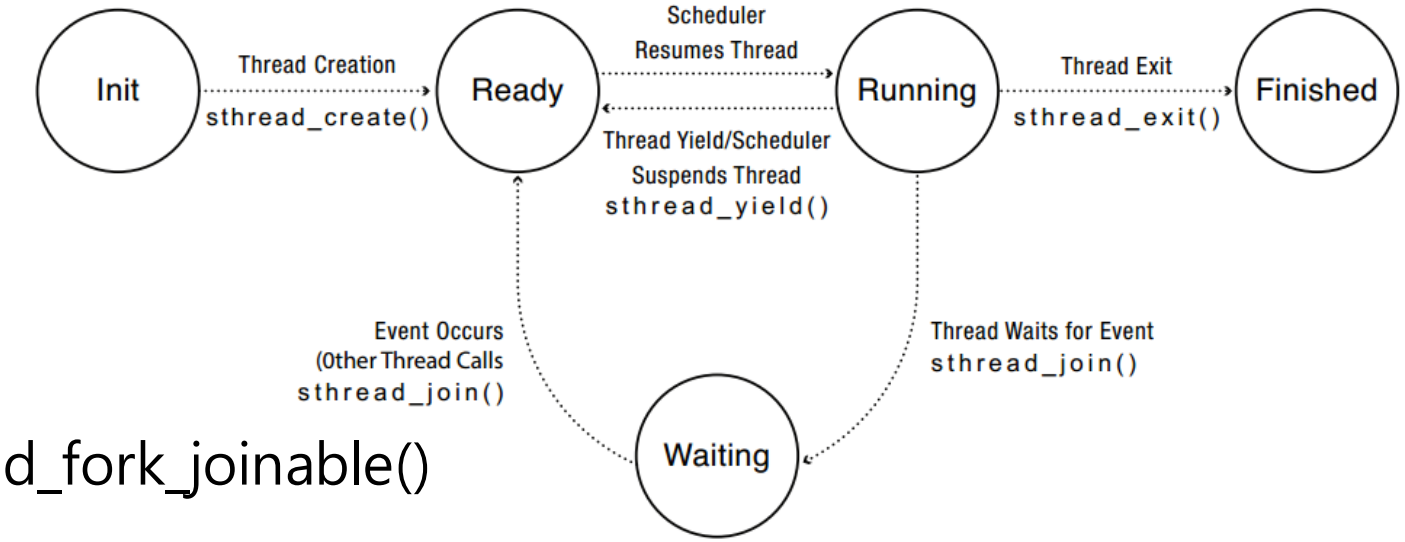
- GDB commands
  - next
  - step
  - break
  - print
  - continue
  - backtrace
  - up
  - down
- http://www.eecs.harvard.edu/~mdw/course/cs161/handouts/gdb.html

Wait Channels

- Use this for implementing other sync primitives in Assignment 1
- Implemented in synch.c
- Protected by a spinlock
  - Caller must hold the spinlock for wchan function call

- wchan_sleep(wc, lk)
  - Put the current thread to sleep and place it in the wchan
  - spinlock is released upon sleep
- wchan_wakeone(wc, lk)
  - Wake up one thread waiting in the wchan
- wchan_wakeall(wc, lk)
  - Wake up all threads waiting in the wchan

Thread Life Cycle



- thread_create
  - Create a new thread
  - We use thread_fork() or thread_fork_joinable()
- thread_yield
  - Relinquish processor
- thread_join
  - **Parent** thread waits for forked thread to exit, then return
  - If the child has already exited, return immediately
- thread_exit
  - Quit thread and clean up, wake up joiner if any
- Garbage collection?
  - What happens if it's not explicitly destroyed?

Thread Join

- Only parent can call it
- Only “joinable” thread created with thread_fork_joinable can b e joined
- Consider cases:
  - The parent joins before the child finishes
  - The parent joins after the child finishes
  - The parent joins while the child is finishing
  - The parent finishes before the child finishes
  - Can the parent join before the child starts?