

Utilities - tmux & git

lctseng (2019-2021, CC BY-SA)

wnlee and others (1996-2018)

國立陽明交通大學資工系資訊中心

Computer Center, Department of Computer Science, NYCU

Tmux

Terminal Multiplexer

國立陽明交通大學資工系資訊中心

Computer Center, Department of Computer Science, NYCU

What is tmux?

- Terminal Multiplexer
 - Allow open multiple tabs (multiple terminals)

The screenshot displays a tmux terminal window with two panes. The left pane shows the vim editor's help screen for the ':help' command. The right pane shows a system status window with a table of running processes and system metrics.

```
VIM - Vi IMproved
      version 7.4.629
  by Bram Moolenaar et al.
Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

  Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version7<Enter> for version info

NORMAL          unix 100% 1:1
```

PID	USER	PRI	NI	VIRT	RES	S	CPU%	MEM%	TIME+	Command
31016	root	20	0	164M	6296	S	3.3	0.0	0:00.05	sshd: ro
30617	fyli	20	0	119M	2252	R	1.3	0.0	0:01.84	htop
31012	fyli	20	0	225M	16652	S	0.7	0.1	0:00.29	vim
820	root	20	0	463M	5424	S	0.7	0.0	4h30:25	NetworkM
31863	hadua	20	0	98M	67316	S	0.7	0.4	5:55.73	(squid-1
756	root	20	0	226M	4064	S	0.7	0.0	2h48:50	vmtoolsd
623	cwchien55	20	0	1137M	29476	S	0.7	0.2	2:01.36	python3
11557	yukc	20	0	21056	1464	S	0.7	0.0	8:20.84	tmux new
1111	root	20	0	1310M	44980	S	0.0	0.3	49h41:21	metricbe
1489	root	20	0	1310M	44980	S	0.0	0.3	3h43:18	metricbe
1153	root	20	0	1310M	44980	S	0.0	0.3	3h35:20	metricbe
1137	root	20	0	1310M	44980	S	0.0	0.3	3h38:40	metricbe
11218	root	20	0	1310M	44980	S	0.0	0.3	3h34:36	metricbe
7050	fyli	20	0	213M	15336	S	0.0	0.1	0:02.57	python3
30090	fyli	20	0	21168	1904	S	0.0	0.0	0:00.57	tmux -2

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice +F
```

```
linux1 fyli ~ ls /
bin boot dev etc home lib lib64 media mnt net opt proc root run sbin srv sys tmp u usr var
linux1 fyli ~
```

```
[0] 0:bash 1:bash- 2:vim* "linux1.cs.nctu.edu.tw" 00:59 09-Sep-20
```

What is tmux?



SSH

Tmux Client

Tmux Server

What is tmux?



Why should I use tmux?

- Keep your working session alive
- By default, shell is **terminated** when connection is lost
 - Including any programs/editors opened
 - Any unsaved changes are **discarded without warning**
- tmux **will not be terminated** when connection is lost
 - Attach to previous sessions!

Example screenshot of tmux

```
78 cat << EOF
79
80 [lctseng]
81
82
83
84
85 EOF
86 echo "^[[1;0m"
87 echo "^[[1;91mATTENTION!!^[[1;0m"
88 echo "Remember to modify ^[[1;91m.gitconfig^[[1;0m to set your own name and email address."
89 echo "Or you can run '^[[1;91mgit config --global user.name \"Your Name\"^[[1;0m' to set your name,"
90 echo "and '^[[1;91mgit config --global user.email you@example.com^[[1;0m' to set your email address"
91
92 username=`whoami`
93 if [ "${username}" != "lctseng" ]; then
94     echo "[Git configuration]"
INSERT  # master setup.sh sh utf-8[unix] 62% 78: 4 [79]tr
-- INSERT --

TMUX(1)      FreeBSD General Commands Manual
NAME
  tmux - terminal multiplexer

SYNOPSIS
  tmux [-2CluvV] [-c shell-command] [-f file]
  [-L socket-name] [-S socket-path] [command [flags]]

DESCRIPTION
  tmux is a terminal multiplexer: it enables a number of terminals to be created, accessed, and controlled from a single screen. tmux may be detached from a screen and continue running in the background, then later reattached.

:
alumni 0-zsh 1-zsh 2-ssh 3-ssh 4-ssh 5-

7:34PM up 190days, 2:56, [0.19, 0.21, 0.19]
```

```
1 [ 2.0%] Tasks: 398, 0 thr; 4 runn
2 [ 0.5%] Load average: 0.19 0.21 0
3 [ 0.0%] Uptime: 190 days(!), 02:5
4 [ 0.6%]
5 [ 0.5%]
6 [ 0.0%]
7 [ 0.5%]
8 [ 0.5%]
9 [ 0.4%]
10 [ 0.0%]
11 [ 0.5%]
12 [ 0.5%]
13 [ 0.0%]
14 [ 2.6%]
15 [ 1.6%]
16 [ 0.5%]
Mem [ 4.68G/16.0G]
Swp [ 1.88G/2.00G]
```

USER	PRI	NI	VIRT	RES	S	CPU%	MEM%	TIME
lctseng	52	0	8844	5056	S	0.0	0.0	0:00.
lctseng	52	0	6944	2924	S	0.0	0.0	0:00.
lctseng	22	0	8972	5984	S	0.0	0.0	0:00.
lctseng	52	0	8972	5980	S	0.0	0.0	0:00.
lctseng	52	0	7148	2984	S	0.0	0.0	0:00.
hlku	20	0	208M	53148	S	0.0	0.3	5h39:
root	-16	0	0	5632	S	0.0	0.0	56h03:
lctseng	20	0	21124	9600	S	0.0	0.1	0:18.
hlku	20	0	108M	44004	S	0.0	0.3	51:20.

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortByF
```

```
[lctseng]
```

<https://github.com/lctseng/Unix-User-Config/blob/master/.tmux.conf>

Advantages of tmux

- Multiple sessions, windows, panes
- Keep the sessions, attach/detach anytime
- Powerful window division (panes)
- Share screen by attaching to the same session

Start tmux

- `tmux`
- `tmux attach [-t <number>]`
- `tmux detach`
- `tmux ls`
- `tmux kill-session [<number>]`

tmux 101

Basic operations and configurations

國立陽明交通大學資工系資訊中心

Computer Center, Department of Computer Science, NYCU

Session

- Create new session (open a new browser)
 - Execute outside of any tmux sessions
 - `$ tmux`
- Detach current session
 - When attached in a session
 - `$ tmux detach`
 - Or close the terminal directly
- Attach to previous session
 - `$ tmux attach`
 - Attach only if previous sessions exist

Multiple sessions (1)

- Open multiple browsers

```
(0) + 0: 4 windows  
(1) + 5: 1 windows  
(2) + 6: 1 windows (attached)  
  
[6] 0:[tmux]* "linux4.cs.nctu.edu.tw" 10:12 09-Sep-20
```


Multiple sessions (2)

- Open multiple browsers
- List opened sessions
 - `$ tmux list-sessions`

```
0: 1 windows (created Sun Jun 16 18:49:57 2019) [128x38]  
1: 3 windows (created Sun Jun 16 18:50:03 2019) [128x38]
```

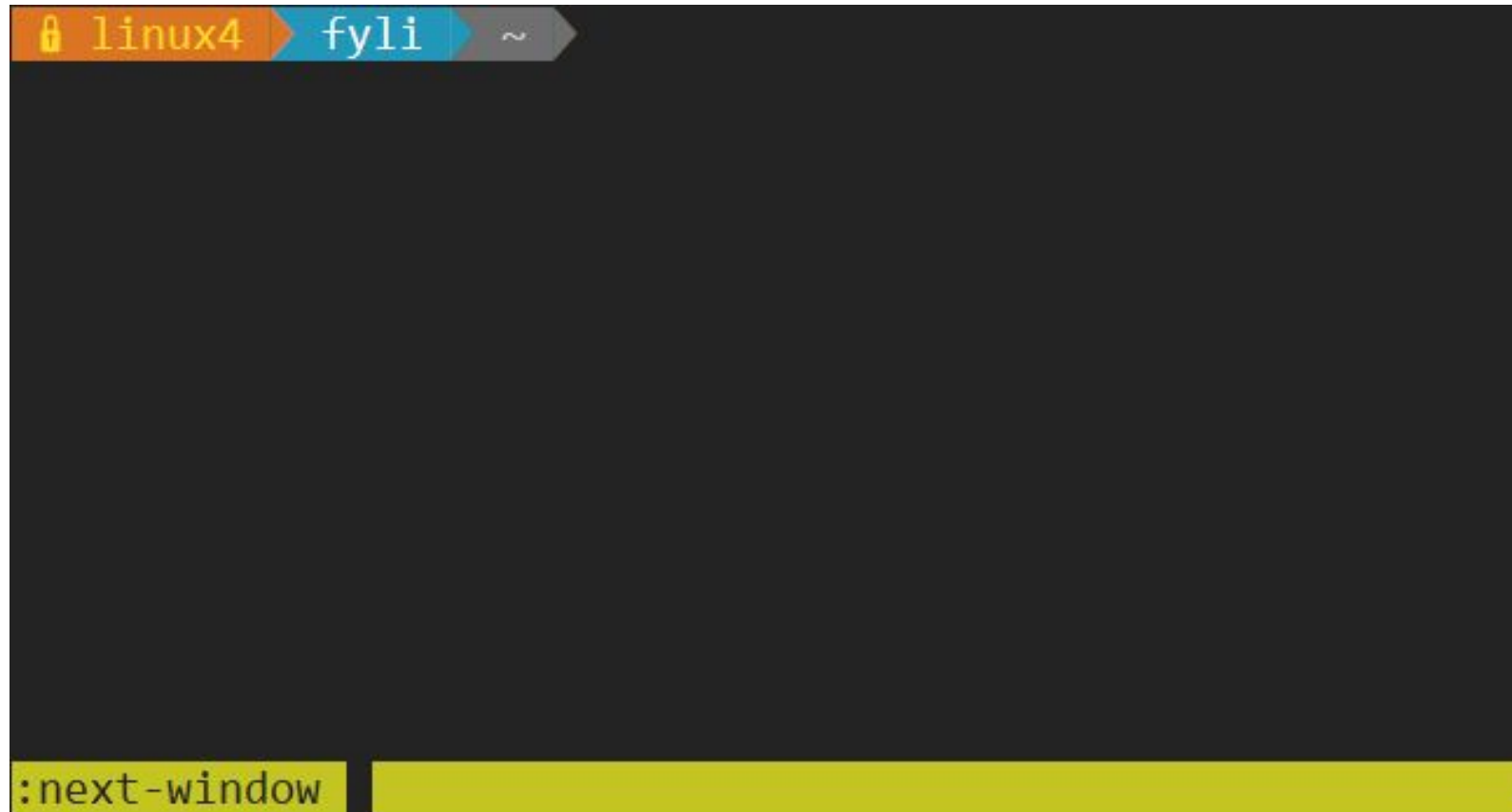
- Attach to previous session by id
 - `$ tmux attach -t session-id`

tmux - bindkey

- Operations start with a special key combination
- Default is C-b
 - Where C is Ctrl (control)

tmux - command

- Bindkey + :
- Open command prompt and execute tmux commands



The image shows a terminal window with a dark background. At the top, there is a header bar with three segments: a lock icon and 'linux4' in orange, 'fyli' in blue, and '~' in grey. The main area of the terminal is empty. At the bottom, there is a yellow bar containing the text ':next-window' in black, indicating that the command prompt is open and ready for input.

Bindkey - Window

bindkey (default is C-b) //C == control	C	new-window
	N	next-window
	P	previous-window
	L	last-window
	<NUM>	select-window -t := <NUM>
	&	confirm-before -p "kill-window #W? (y/n)" kill-window
	,	command-prompt -I "#W" "rename-window '%0%0'"
	.	command-prompt "move-window -t '%0%0'"

Bindkey - Pane

bindkey (default is C-b) //C == control	%	split-window -h
	“	split-window
	arrow-key	select-pane
	alt + arrow-kwy	resize-pane
	x	confirm-before -p "kill-pane #P? (y/n)" kill-pane
	{	swap-pane -U
	}	swap-pane -D

tmux - bindkey

- bindkey + ?

```
bind-key      C-b send-prefix      [57/57]
bind-key      C-o rotate-window
bind-key      C-z suspend-client
bind-key      Space next-layout
bind-key      ! break-pane
bind-key      " split-window
bind-key      # list-buffers
bind-key      $ command-prompt -I #S "rename-session '"
bind-key      % split-window -h
bind-key      & confirm-before -p "kill-window #W? (y/w
bind-key      ' command-prompt -p index "select-window"
bind-key      ( switch-client -p
bind-key      ) switch-client -n
bind-key      , command-prompt -I #W "rename-window '%"
bind-key      - delete-buffer
bind-key      . command-prompt "move-window -t '%"'"
[6] 0:[tmux]* "linux4.cs.nctu.edu.tw" 10:34 09-Sep-20
```

Configuration - tmux.conf

- `~/.tmux.conf`
- design yourself style
- colorful

Configuration - tmux.conf

- ~/.tmux.conf

```
1 set -g status-utf8 on
2 setw -g utf8 on
3 # GENERAL SETTING
4 bind-key r source-file ~/.tmux.conf; display-message "~/.tmux.conf is reloaded"
5 set-window-option -g automatic-rename off
6 set-option -g default-terminal "xterm"
7 set-option -g prefix C-a
8
9 # STATUSBAR STYLE
10 # main
11 set-option -g status-bg colour236
12 set-option -g status-fg colour166
13 setw -g window-status-current-format "#I:#W#F"
14 setw -g window-status-current-fg colour215
15
16 #left
17 set-option -g status-left ''
18 set-option -g status-left-length 0
19
20 #right
21 set-option -g status-right "#h [%Y-%m-%d %H:%M]"
22 # BIND KEY
23 bind -n F8 previous-window
24 bind -n F9 next-window
25 bind -n F10 last-window
26 bind -n M-Right next-window
27 bind -n M-Left previous-window

"~/.tmux.conf" 27L, 698C
0:tbsd21 1:csduty* 2:cshome 3:csschedule- 4:bsd3 5:jail 6:nasa 7:jsh
```


Configuration - bindkey

- ~/.tmux.conf
 - bind-key (alias: bind)
 - C (alias: <Ctrl>)
 - M (alias: <Alt>)
- bind-key <key> <command>
 - -T key-table (default table is prefix)
 - -n : alias for -T root => **Don't need to press C-b first**
 - -r : repeat

```
22 # BIND KEY
23 bind -n F8 previous-window
24 bind -n F9 next-window
25 bind -n F10 last-window
26 bind -n M-Right next-window
27 bind -n M-Left previous-window
```

Configuration - set

- ~/.tmux.conf
 - set-window-option (alias: setw)

```
1 set -g status-utf8 on
2 setw -g utf8 on
3 # GENERAL SETTING
4 bind-key r source-file ~/.tmux.conf; display-message "~/.tmux.conf is reloaded"
5 set-window-option -g automatic-rename off
6 set-option -g default-terminal "xterm"
7 set-option -g prefix C-a
8
9 # STATUSBAR STYLE
10 # main
11 set-option -g status-bg colour236
12 set-option -g status-fg colour166
13 setw -g window-status-current-format "#I:#W#F"
14 setw -g window-status-current-fg colour215
15
16 #left
17 set-option -g status-left ''
18 set-option -g status-left-length 0
19
20 #right
21 set-option -g status-right "#h [%Y-%m-%d %H:%M]"
```


Configuration - set

colour8	colour1	colour2	colour3	colour4	colour5	colour6	colour7
	colour9	colour10	colour11	colour12	colour13	colour14	colour15
	colour17	colour18	colour19	colour20	colour21	colour22	colour23
colour24	colour25	colour26	colour27	colour28	colour29	colour30	colour31
colour32	colour33	colour34	colour35	colour36	colour37	colour38	colour39
colour40	colour41	colour42	colour43	colour44	colour45	colour46	colour47
colour48	colour49	colour50	colour51	colour52	colour53	colour54	colour55
colour56	colour57	colour58	colour59	colour60	colour61	colour62	colour63
colour64	colour65	colour66	colour67	colour68	colour69	colour70	colour71
colour72	colour73	colour74	colour75	colour76	colour77	colour78	colour79
colour80	colour81	colour82	colour83	colour84	colour85	colour86	colour87
colour88	colour89	colour90	colour91	colour92	colour93	colour94	colour95
colour96	colour97	colour98	colour99	colour100	colour101	colour102	colour103
colour104	colour105	colour106	colour107	colour108	colour109	colour110	colour111
colour112	colour113	colour114	colour115	colour116	colour117	colour118	colour119
colour120	colour121	colour122	colour123	colour124	colour125	colour126	colour127
colour128	colour129	colour130	colour131	colour132	colour133	colour134	colour135
colour136	colour137	colour138	colour139	colour140	colour141	colour142	colour143
colour144	colour145	colour146	colour147	colour148	colour149	colour150	colour151
colour152	colour153	colour154	colour155	colour156	colour157	colour158	colour159
colour160	colour161	colour162	colour163	colour164	colour165	colour166	colour167
colour168	colour169	colour170	colour171	colour172	colour173	colour174	colour175
colour176	colour177	colour178	colour179	colour180	colour181	colour182	colour183
colour184	colour185	colour186	colour187	colour188	colour189	colour190	colour191
colour192	colour193	colour194	colour195	colour196	colour197	colour198	colour199
colour200	colour201	colour202	colour203	colour204	colour205	colour206	colour207
colour208	colour209	colour210	colour211	colour212	colour213	colour214	colour215
colour216	colour217	colour218	colour219	colour220	colour221	colour222	colour223
colour224	colour225	colour226	colour227	colour228	colour229	colour230	colour231
	colour233	colour234	colour235	colour236	colour237	colour238	colour239
colour240	colour241	colour242	colour243	colour244	colour245	colour246	colour247
colour248	colour249	colour250	colour251	colour252	colour253	colour254	colour255

tmux - share session

- Both side can edit and execute commands

```
linux4.cs.nctu.edu.tw - PuTTY
linux4 fyli ~ curl -4 ifconfig.co
140.113.235.154
linux4 fyli ~ whoami
fyli
linux4 fyli ~
[8] <sh* "linux4.cs.nctu.edu.tw" 03:23 14-Sep-20
```

```
linux4.cs.nctu.edu.tw - PuTTY
linux4 fyli ~ curl -4 ifconfig.co
140.113.235.154
linux4 fyli ~ whoami
fyli
linux4 fyli ~
[8] <sh* "linux4.cs.nctu.edu.tw" 03:23 14-Sep-20
```


Reference

- [tmux \(1\)](#)
- tmux shortcuts & cheatsheet
 - <https://gist.github.com/MohamedAlaa/2961058>
- tmux brief introduction (chinese)
 - <https://5xruby.tw/posts/tmux/>

Appendix: tmux v.s. screen

	tmux	screen
Window Split	top-down & left-right	top-down only (default)
	different sessions can have different schemes	scheme must be shared by all sessions
Session	switch between sessions without detach	detach first then re-attach another session
	multiple clients can attach to same session	one session for one client only (force detach)
Profile	.tmux	.screenrc
	highly customizable	less than tmux

Git

One of the most popular Version Control Systems

國立陽明交通大學資工系資訊中心

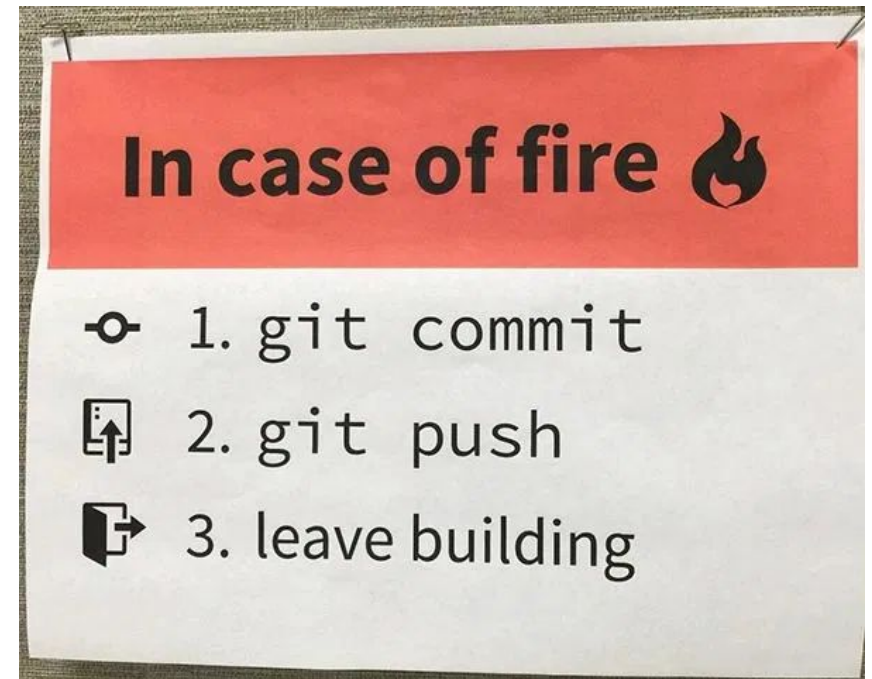
Computer Center, Department of Computer Science, NYCU

Version Control Systems (VCS)

- Also known as Source Code Management (SCM)
- Records changes to a set of files over time so that you can recall specific versions later.
- Easy for developing, finding bug, blame someone else, ...
- Popular tools
 - Git, Subversion (svn), Mercurial (hg)

Version Control Systems (VCS)

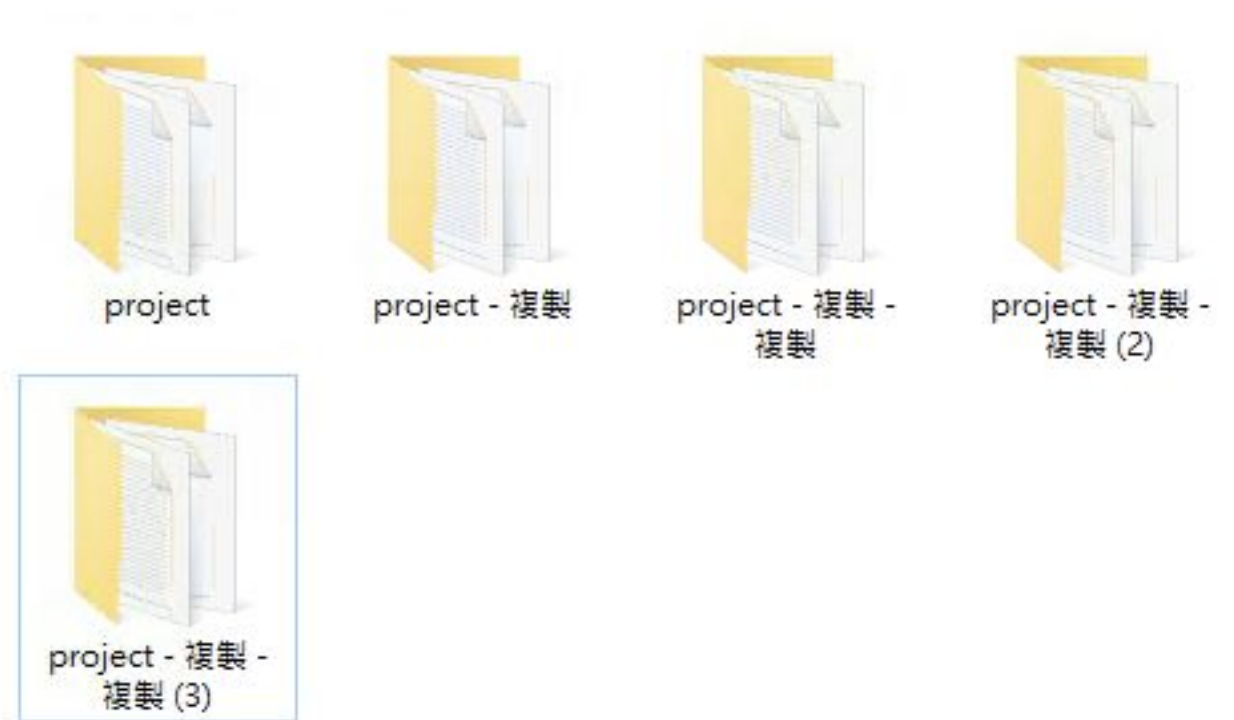
- Web hosting
 - Backup projects
 - Collaborating
 - Commercial providers
 - Github, Gitlab, Bitbucket
 - Self-hosted
 - `git.cs.nctu.edu.tw`



Without VCS

- Copy-paste manually

```
# 2015-11-10  
cp -r project project.bak  
# 2015-11-11  
cp -r project project.bak1  
# 2015-11-12  
cp -r project project.bak2  
# 2015-11-15  
cp -r project project.bak3
```

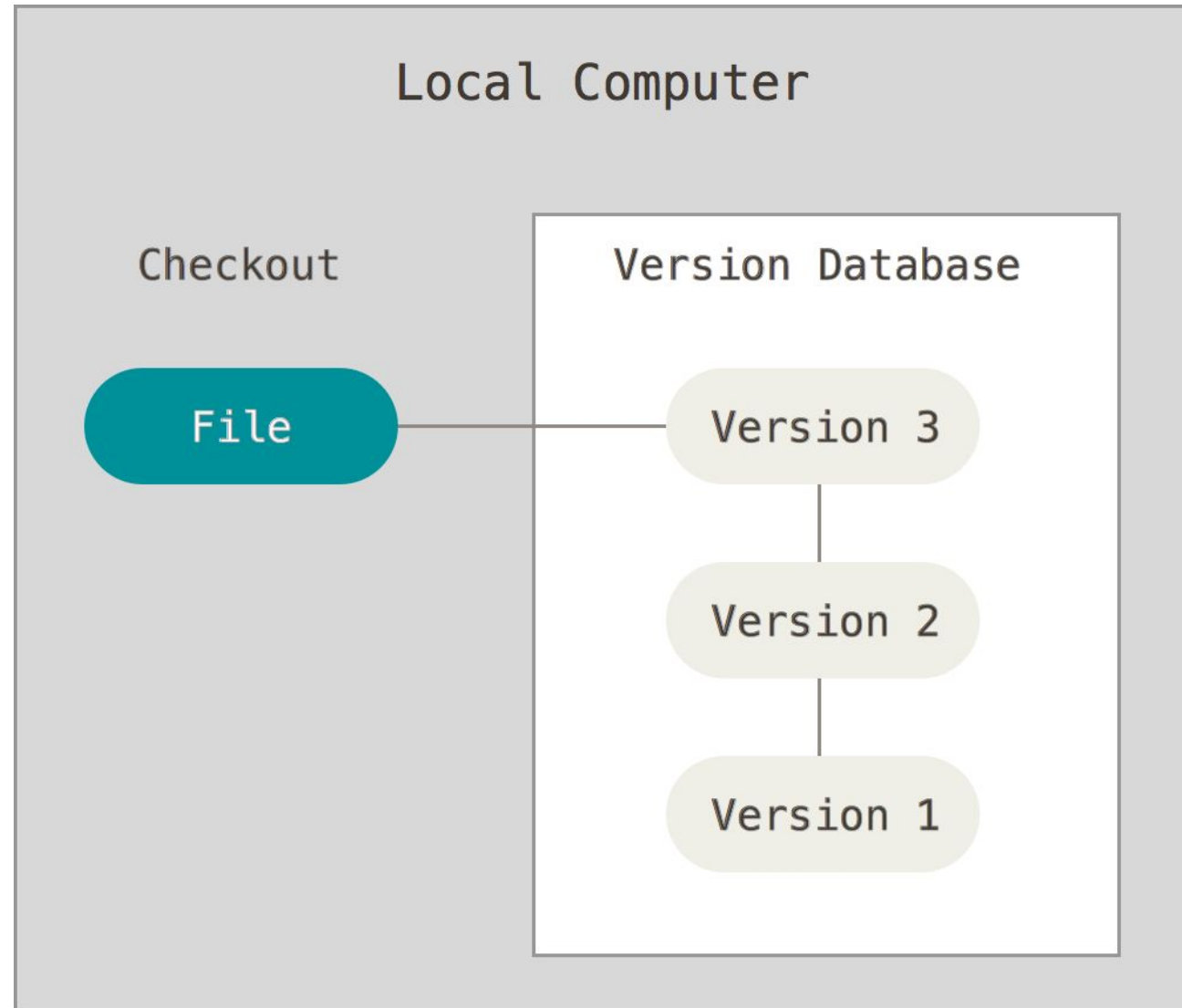


How VCS works

- In addition to your files, VCS stores extra information under your project folder
 - Hidden folders
 - .git for Git
 - .hg for Mercurial
 - Previous versions of files (compressed)
 - Remote repository information

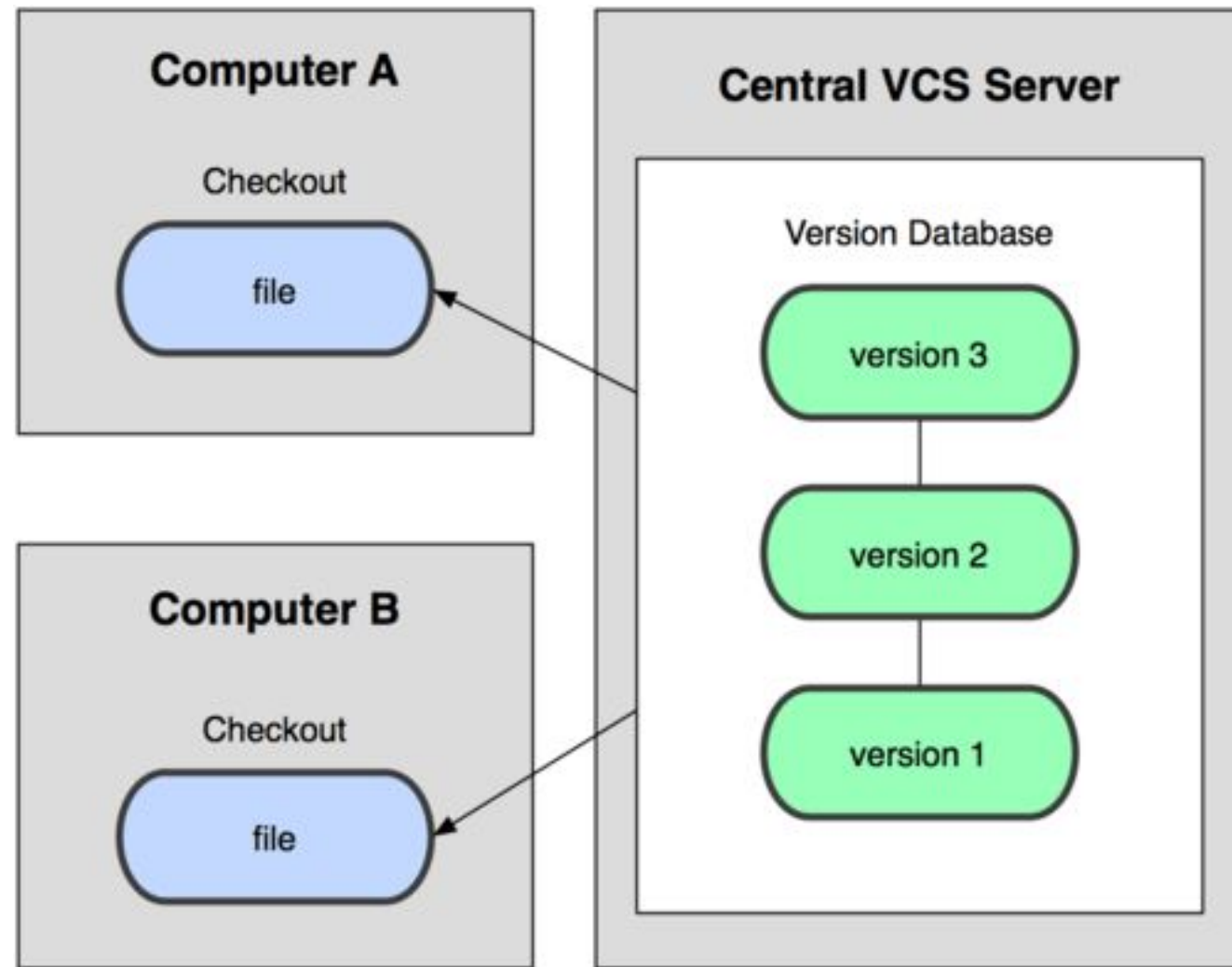
Types of VCS (1)

- Local VCS
 - All versions are in local
 - Cannot share with others
 - No remote backup
- Example
 - Manually copy-paste
 - Git/Hg without setting remote upstream



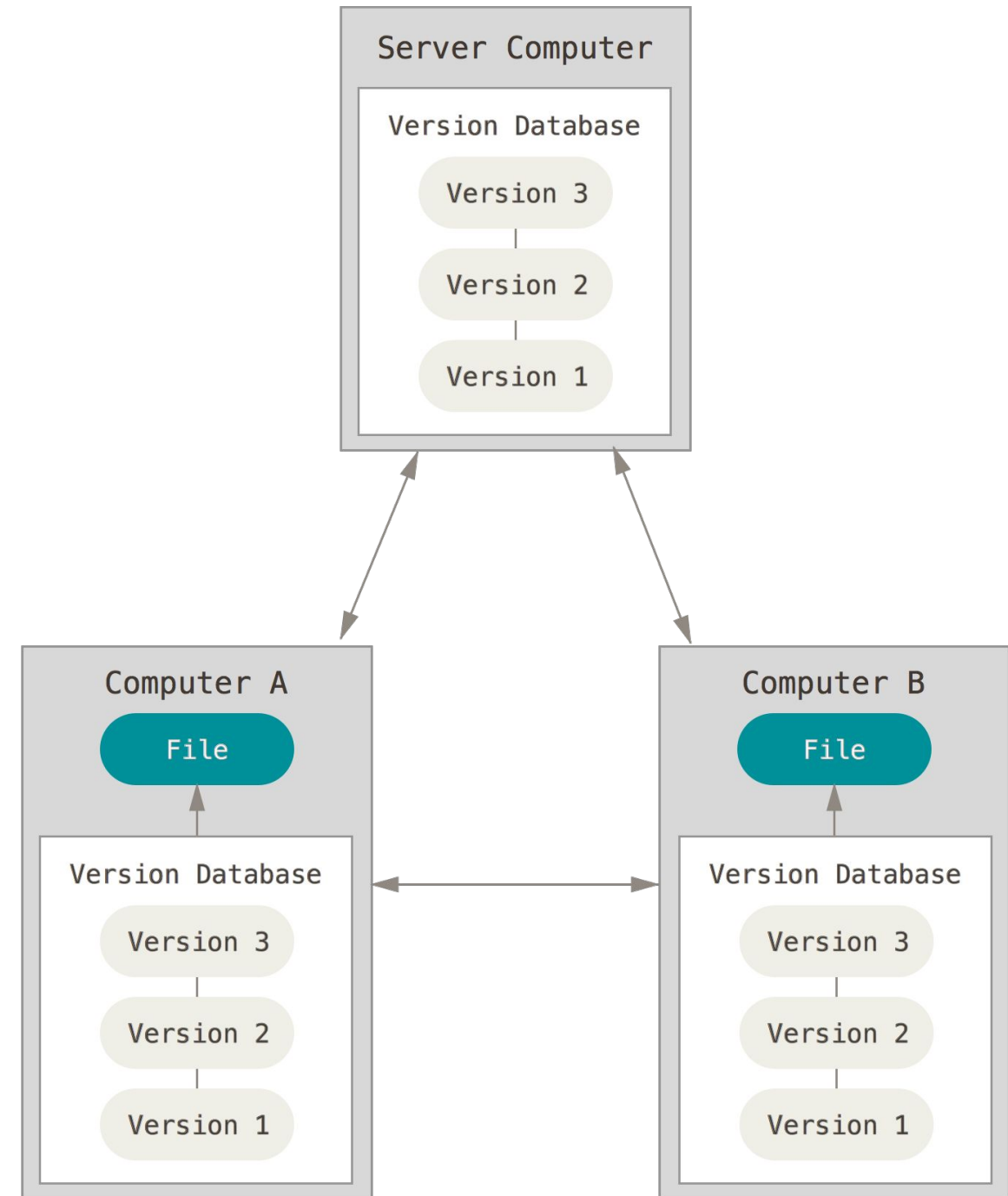
Types of VCS (2)

- Centralized VCS
 - User can checkout one specific version
 - Remote server has all versions
 - Lost access to other versions if network is down
 - Lost all versions if server is down
- Example
 - Subversion



Types of VCS (3)

- Distributed VCS
 - Every node has complete copy of versions
 - Offline working
 - Synchronization
 - Usually using a server as the source of truth
- Example
 - Git, Mercurial



Git

- Distributed VCS
- History
 - Linus Torvalds (the creator of Linux)
 - Source Control Management for Linux Kernel
 - ~2005
 - Using BitKeeper (commercial software)
 - 2005~
 - Developed Git

Git

- Snapshots, not differences
 - All versions of all files are stored independently
 - Easy for checking out to any version
- Nearly every operation is local
- Git has integrity
 - SHA checking
- Git generally only adds data
 - Delete
 - Store the file in `.git` folder and hide from your workspace
 - Modify
 - Backup the original file to `.git` folder

Git

- Git is very powerful, with many features
- In this class, we only talk about the very simple one
- We will cover
 - How to install and create repos
 - How to add files/make changes
 - How to create commits
 - How to navigate between versions
 - How to push to remote
- Will NOT cover
 - branching (git branch, git merge, ...)
 - Anything else

Homework 2

Git - installation

- FreeBSD
 - pkg install git
- Other OS
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

```
> git
usage: git [--version] [--help] [-C <path>] [-c name=value]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          <command> [<args>]

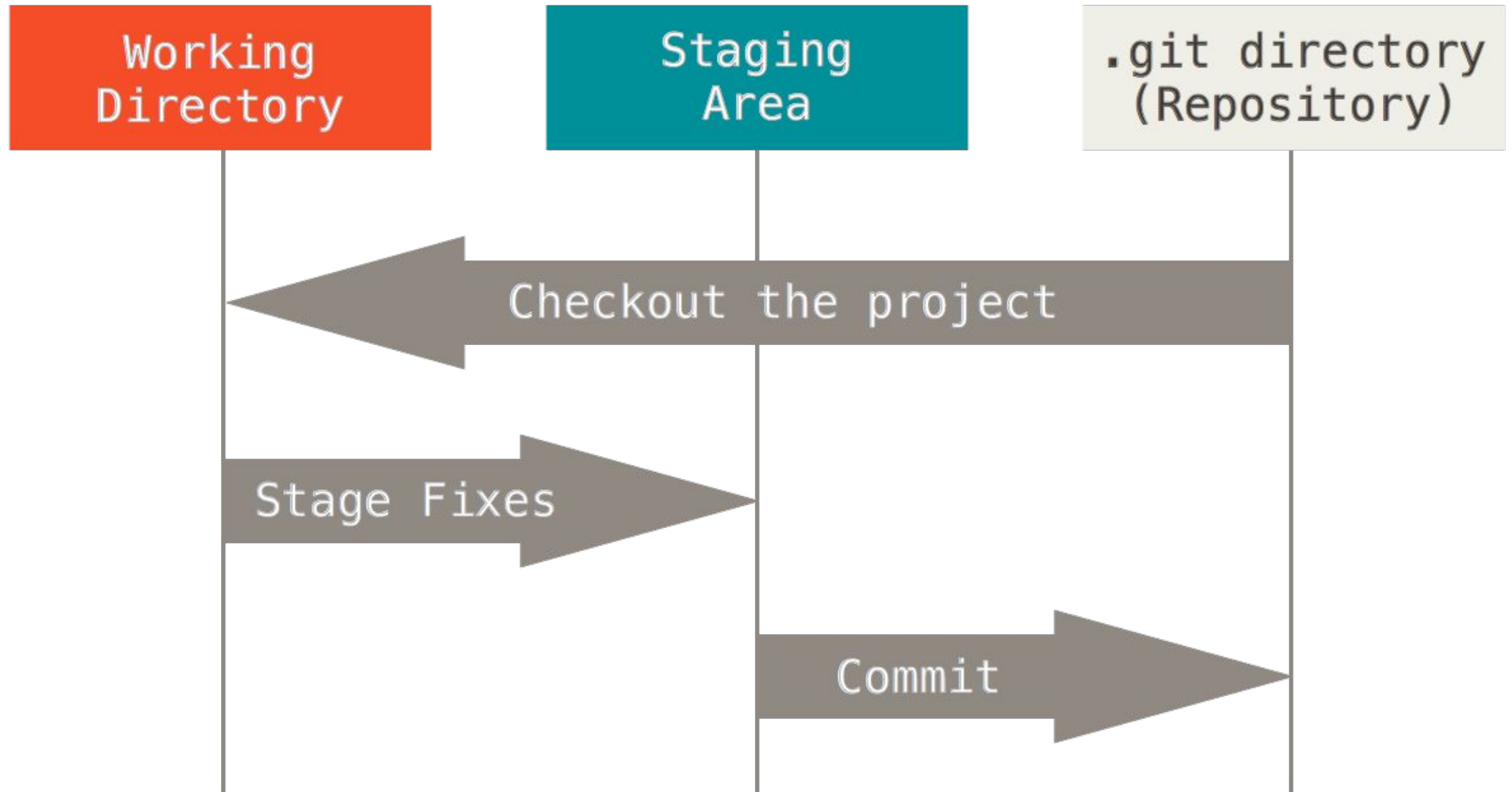
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one
```

Git - getting started

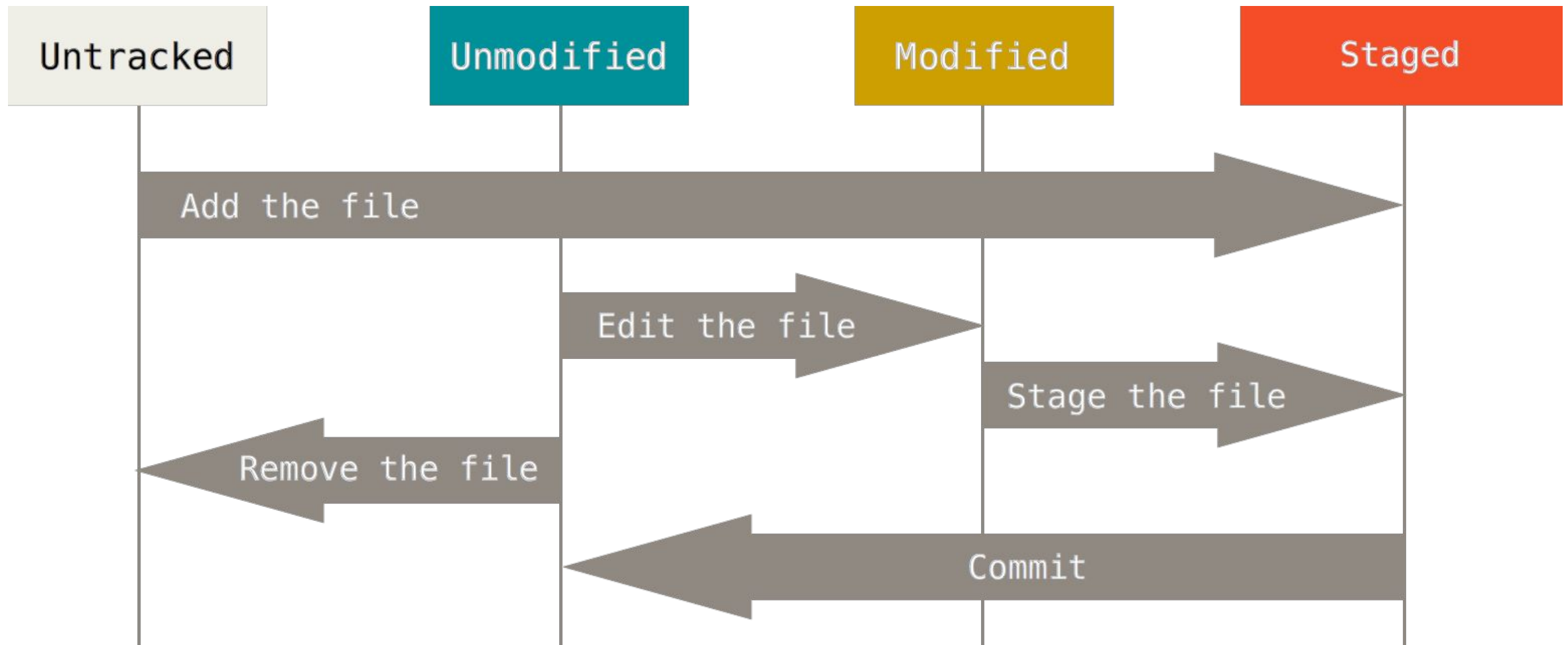
- Developer information
 - In terminal
 - `$ git config --global user.email "you@example.com"`
 - `$ git config --global user.name "Name"`
 - Or edit manually in `~/.gitconfig`
- Create a new repository
 - `$ mkdir hw2`
 - `$ cd hw2`
 - `$ git init`
 - Initialize an empty project with `.git` directory

Git - the three stages



Git - file lifecycle

- The lifecycle of the status of your files



Git - basic operations (1)

- git add
 - Add file contents to the index (staged)
 - `$ git add file1 file2...`
 - `$ git add directory`
 - `$ git add .`

Git - basic operations (2)

- `git commit`
 - Confirm your staged change and create a commit (revision)
 - Will open default text editor for commit message
 - Useful options
 - `-a`: stage all modified and deleted path (`git add`)
 - `-m MSG`: use `MSG` as commit message without opening editors

Git - basic operations (3)

- git status
 - View the working directory status of current project

```
$ git status -s
M README.md          # updated in index
D run.sh             # deleted from index
R src/main.js        # renamed in index
A src/index.html     # added to index
?? src/README.md     # untracked
```

Git - basic operations (4)

- git diff
 - Compare change with the previous version
 - <https://git-scm.com/docs/git-diff>

```
$ git diff
diff --git a/README.md b/README.md
index 76f177f..f4986c2 100644
--- a/README.md
+++ b/README.md
@@ -1, +1 @@
-# Hi
+# Hello
```


Git - basic operations (5)

- git log
 - View the commit log (history)
 - Change the log format
 - <http://gits-scm.com/docs/pretty-formats>

```
$ git log

commit 2f2bd00051fbd4d4978b7e96508b97950d6e60b1
Author: lctseng lctseng@cs.nctu.edu.tw
Date:   Thu Oct 8 17:10:13 2020 +0800

    Initial commit
```

Git - basic operations (6)

- git checkout
 - Go to other revisions or branches
 - `$ git checkout <commit-id>`
 - Go back to latest change:
 - `$ git checkout master`

```
$ git status
On branch master

$ git checkout 2f2bd00
Note: checking out '2f2bd00'.

$ git status
HEAD detached at 2f2bd00
```

Git - basic operations (7)

- git bisect
 - Finding problematic commit
 - Flow (similar to binary search)
 - `$ git bisect start`
 - Define endpoints
 - `$ git bisect good <good-commit-id>`
 - `$ git bisect bad <bad-commit-id>`
 - After that, git will checkout to some commit
 - If that commit is good
 - `$ git bisect good`
 - If that commit is bad
 - `$ git bisect bad`
 - After several rounds, git will checkout to first bad commit

Git - push to remote

- git remote
 - Manage the remote repository
 - `$ git remote add origin`
<https://git.cs.nctu.edu.tw/lctseng/sa-demo.git>
 - Set the remote server & repo for push/pull
- git push
 - `$ git push -u origin master`
 - Push to the remote and create the remote master branch
 - Later on, you can simply use "git push"

Git - pull from remote

- git clone
 - Download the full repository from remote server
 - `$ git clone https://github.com/curl/curl.git`
- git pull
 - Pull the latest revisions from remote in an existing repository