

CH3 GNU Software



GNU Software

- Open Source License

- Develop Tool

 - Geany

 - diff, patch

 - Git

- Build Code

 - Makefile

Open Source License

➤ GNU General Public License

- 只要在一個軟件中使用 (" 使用 " 指類庫引用，修改後的代碼或者衍生代碼) GPL 協議的產品，則該軟件產品必須也採用 GPL 協議，既必須也是開源和免費。這就是所謂的 " 傳染性 "

➤ BSD License

- 基本上使用者可以 " 為所欲為 "，可以自由的使用，修改源代碼，也可以將修改後的代碼作為開源或者專有軟件再發佈。

➤ LGPL

- LGPL 是 GPL 的一個為主要為類庫使用設計的開源協議。LGPL 允許商業軟件通過類庫引用 (link) 方式使用 LGPL 類庫而不需要開源商業軟件的代碼。這使得採用 LGPL 協議的開源代碼可以被商業軟件作為類庫引用並發布和銷售。

Develop Tool



Geany

➤ Geany

➤ <https://www.geany.org/>

➤ `sudo apt-get install geany`



Tracking code command

» Linux command

» Find -name "*.c" | xargs grep -n "function name"

diff and patch

➤ diff - compare files line by line

➤ Create a patch file

- `diff -Nuar file_a file_b > c.patch`
 - `-N`, treat absent files as empty
 - `-a`, `--text`
 - `-u`, output NUM (default 3) lines of unified context
 - `-r`, recursively compare any subdirectories found

➤ patch - apply a diff file to an original

➤ apply a patch file

- `patch ./hello_1.c < ./tmp.patch`

➤ Reverse a patch file

- `patch -R ./hello_1.c < tmp.patch`



Git

➤ <https://git-scm.com/book/zh-tw/v1/>

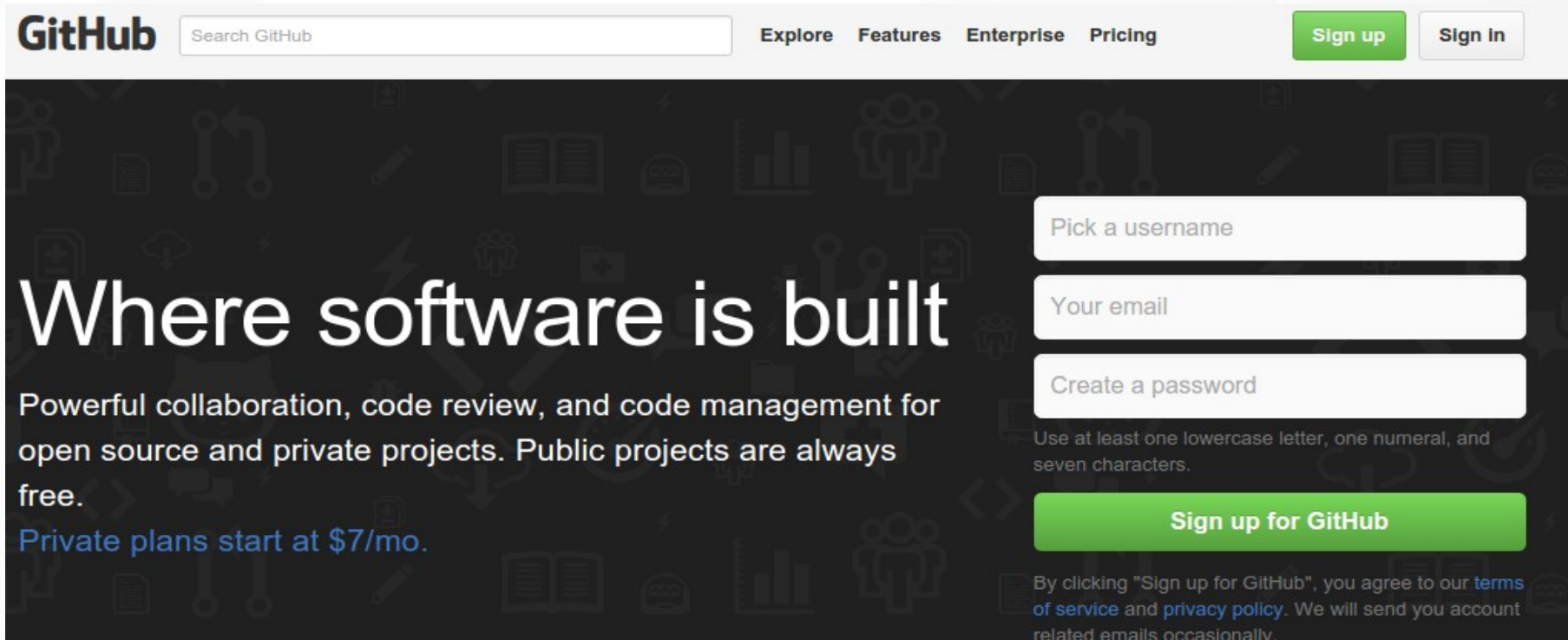
➤ 版本控制

➤ 程式回溯

➤ 管理多人共同開發

GitHub

 <https://github.com/>



The screenshot shows the GitHub homepage with a dark background and various icons. The main heading is "Where software is built". Below it, a paragraph describes the platform's capabilities for collaboration, code review, and code management. A link for private plans is provided. On the right side, there is a sign-up form with three input fields: "Pick a username", "Your email", and "Create a password". A green "Sign up for GitHub" button is below the form. At the bottom right, there is a disclaimer about agreeing to terms of service and privacy policy.

GitHub Search GitHub

Explore Features Enterprise Pricing

[Sign up](#) [Sign In](#)

Where software is built

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free.

Private plans start at \$7/mo.

Pick a username

Your email

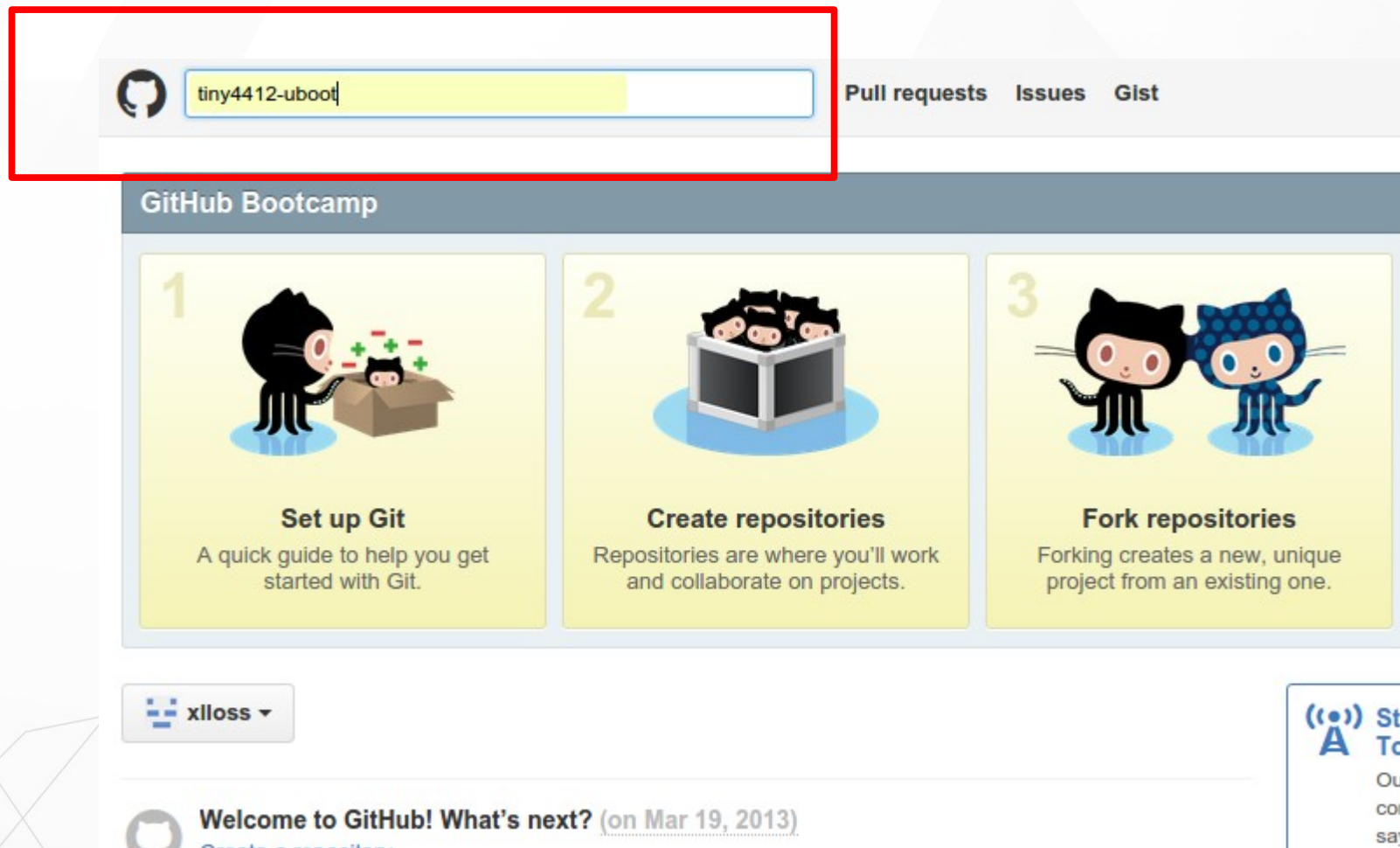
Create a password

Use at least one lowercase letter, one numeral, and seven characters.

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

GitHub



GitHub

xlloss / **tiny4412-uboot**
forked from embeddedclass/tiny4412-uboot

Unwatch 1 Star 0 Fork 1

Description

Short description of this repository

Website

Website for this repository (optional)

Save or Cancel

2 commits

1 branch

0 releases

1 contributor



Branch: master

tiny4412-uboot / +



This branch is even with embeddedclass:master.

Pull request Compare



Slash.Huang tiny 4412 u-boot start up

Latest commit 7f37022 an hour ago



uboot_tiny4412

tiny 4412 u-boot start up

an hour ago



README.md

Create README.md

3 months ago

README.md

tiny4412-uboot

<> Code

Pull requests 0

Wiki

Pulse

Graphs

Settings

HTTPS clone URL

https://github.c



You can clone with [HTTPS](#),
[SSH](#), or [Subversion](#).

Download ZIP

Exercise

➤ 1. Clone code to local

➤ #git clone https://github.com/xlloss/tiny4412-uboot.git

➤ 2. modify something

➤ #gedit README

➤ 3. check source status

➤ #git status

➤ 4. use "git add <file>..." to update what will be committed

➤ git add ./README

➤ 5. check status again

➤ git status

Exercise

➤ 6. commit code to local repository

➤ #git commit -a "test"

➤ Or #git commit

➤ 7. check log

➤ #git log

➤ 8. check how many branch in local repository

➤ #git branch

➤ 9. create new branch in local repository

➤ #git branch "new_branch_name"

➤ #git branch slash_uboot

Exercise

➤ 10. check out to new branch

➤ #git checkout "branch_name"

➤ #git checkout slash_uboot

➤ 11. check branch again

➤ #git branch

➤ 12 . push log branch to remote

➤ git push origin slash-uboot

➤ 13. check remote branch status

➤ #git branch origin/ and push tab x2

Exercise

➤ reset your code, but modify code still live

➤ #git reset commit hash coed

➤ Hard reset your code, all modify code will discard

➤ #git reset - -hard hash coed

➤ Check log

➤ #git log

➤ #git show

➤ Download objects and refs from another repository

➤ #git fetch [--all]



BASIC Git Command

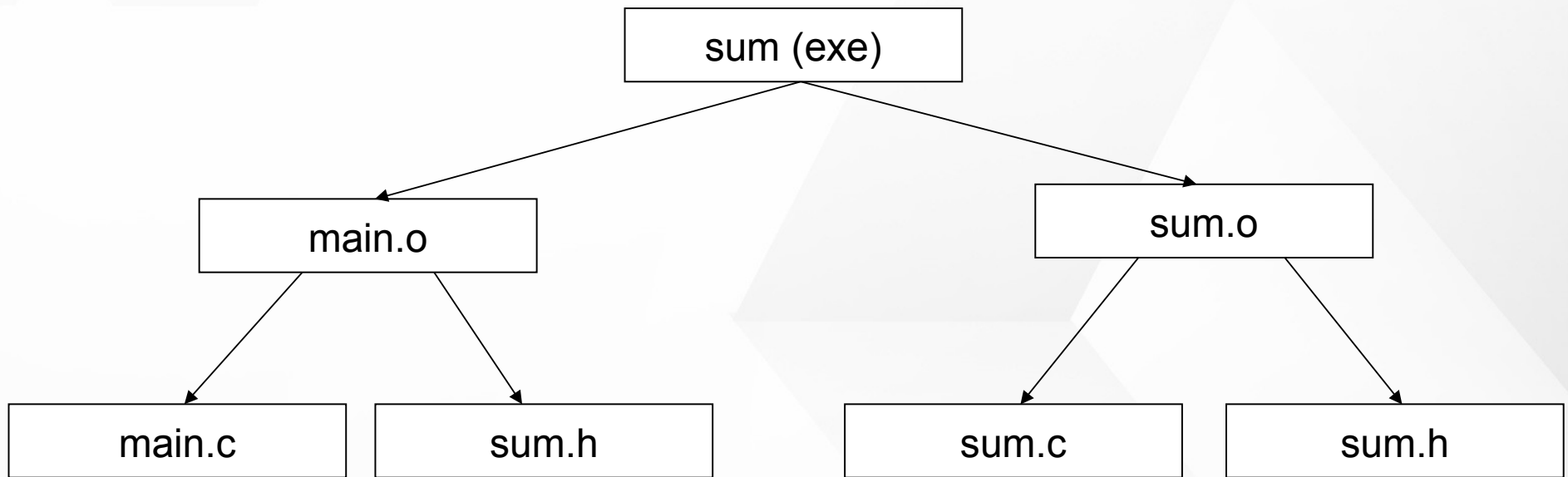
- **add** Add file contents to the index
- **branch** List, create, or delete branches
- **checkout** Checkout a branch or paths to the working tree
- **clone** Clone a repository into a new directory
- **commit** Record changes to the repository
- **diff** Show changes between commits, commit and working tree, etc
- **rm** Remove files from the working tree and from the index
- **pull** Fetch from and merge with another repository or a local branch
- **push** Update remote refs along with associated objects
- **reset** Reset current HEAD to the specified state
- **cherry-pick** apply changes introduced by some existing commits



Makefile

- Simplify compile command
- Automation compile, linker program source
- It can update source in accordance with the dependence

Makefile





Makefile

```
sum: main.o sum.o
```

```
gcc -o sum main.o sum.o
```

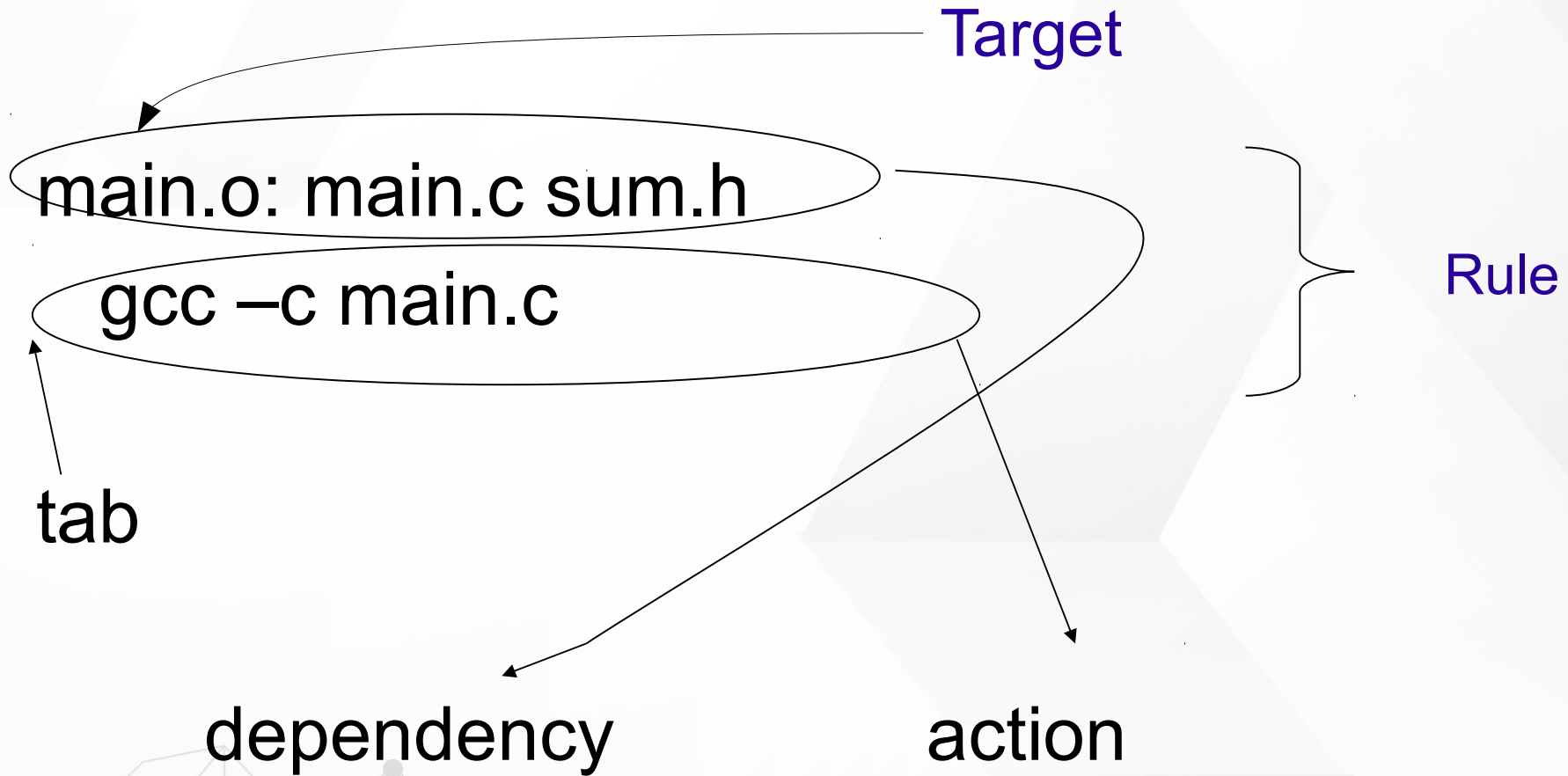
```
main.o: main.c sum.h
```

```
gcc -c main.c
```

```
sum.o: sum.c sum.h
```

```
gcc -c sum.c
```

Rule syntax



Assignment Operators

➤ =

➤ Defines a recursively expanded variable

➤ :=

➤ Defines a simply expanded variable

➤ +=

➤ Also called the append operator. Appends more characters to the existing value of a variable

➤ ?=

➤ The conditional assignment operator. Assigns a value to a variable, but only if the variable has no value, otherwise keep original value

Command-Line Options

➤ -C dir, --directory= dir

➤ make changes the current working directory to dir before it does anything else. If the command line includes multiple -C options, each directory specified builds on the previous one

➤ -I dir, --include-dir= dir

➤ If a makefile contains include directives that specify files without absolute paths, search for such files in the directory.

➤ -j [number] , --jobs[= number]

➤ Run multiple commands in parallel