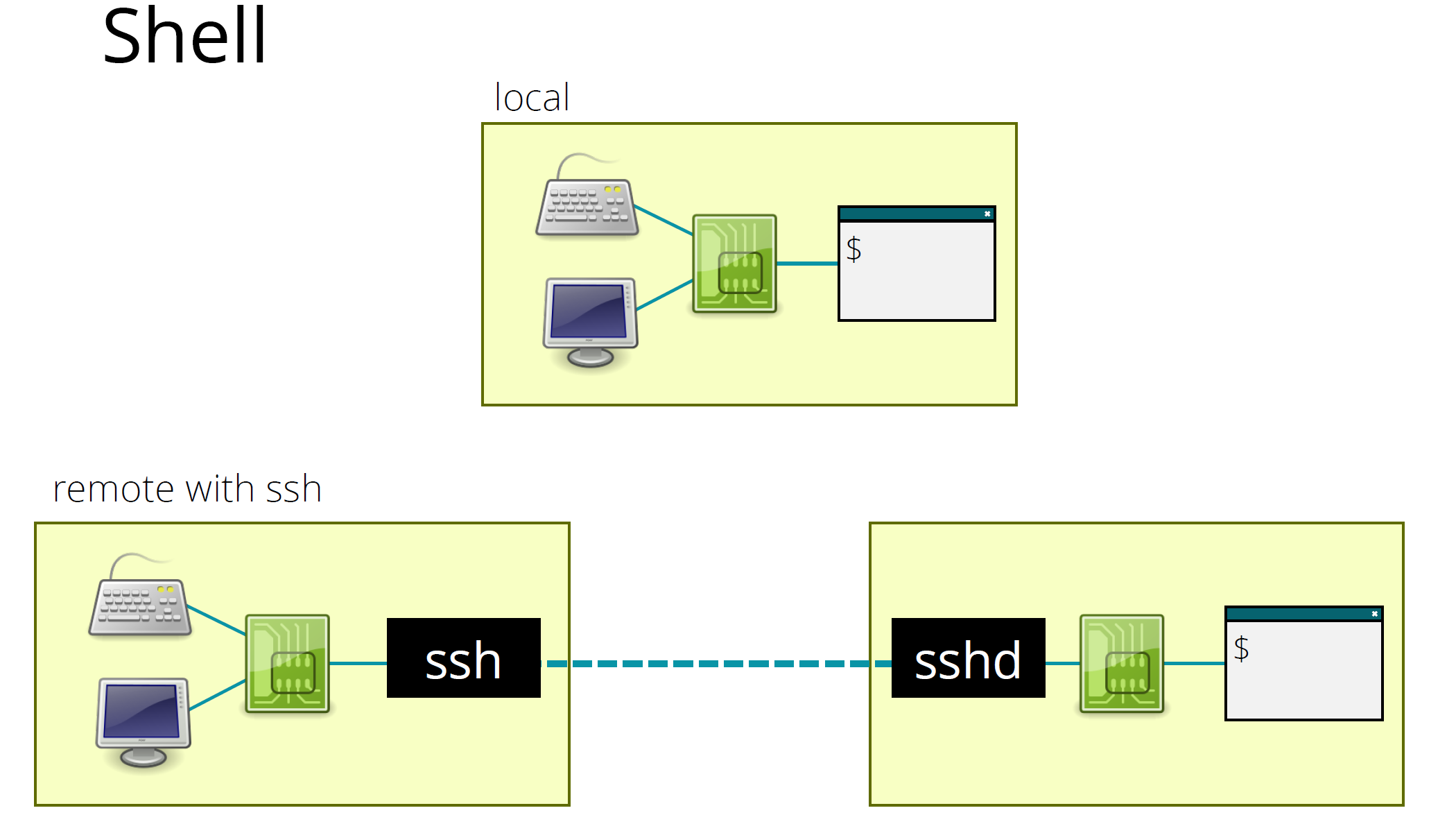
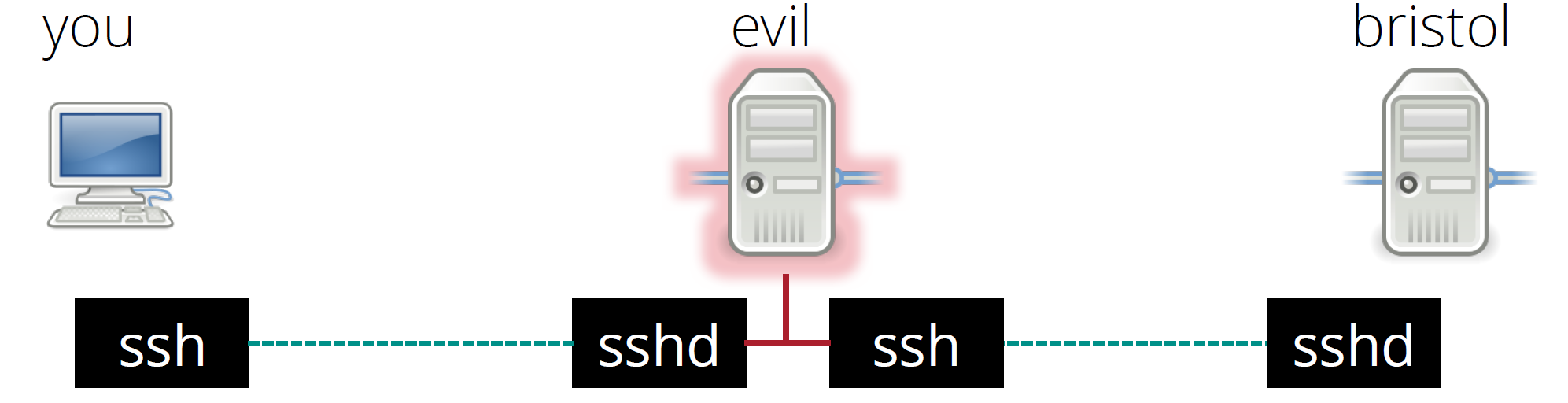
1. **Sysadmin**

**** ****

**ssh [USERNAME@]HOSTNAME**

**Ssh**: Secure shell（SSH）是一种允许您远程登录到另一台计算机（如实验室机器）的协议，如OpenSSH。SSH是客户机，您可以在自己的机器上运行它来连接到另一台机器。

sshd是服务器，在unix中是守护进程（daemon）。它在您想要连接的机器的后台运行，并且需要由系统管理员安装。注意：SSH默认使用TCP端口22。

1. SSH servers have a key pair for digital signatures. If you connect to a new server, SSH asks you if you're sure. If you connect to a server and the key has changed, you get a warning.

**Pwd** print working directory

[USERNAME@]HOSTNAME @前后永远是username和hostname

~ 即home directory /home/username

./ 代表当前directory

**检查ssh客户端是否正常工作：**

ssh localhost：

-要求输入密码：ssh客户端正在工作，并且ssh服务器正在您当前的机器上运行

-成功：客户机正在工作，并且ssh服务器正在您的机器上运行，并且您没有密码

-connection refused：ssh客户端正常工作，但您自己的机器上没有运行服务器

-error：you don't have (Open)SSH installed

**连接lab**

ssh [USERNAME@]HOSTNAME

ssh [user@seis.bris.ac.uk](mailto:user@seis.bris.ac.uk) | SEIS: Not a lab machine, but accessible from the internet. 您想要使用的大多数软件（如编译器）都没有安装在那里。但是，在seis上确实有一个主目录，用于存储SSH密钥之类的东西。bastion host, accessible from internet, but doesn't have your files or programs

ssh [user@rd-mvb-linuxlab.bristol.ac.uk](mailto:user@rd-mvb-linuxlab.bristol.ac.uk) | load-balancer randomly assigns to a lab machine!

it######.wks.bris.ac.uk | lab machines, #=075637 to 075912

或使用ssh -J USERNAME@seis.bris.ac.uk [USERNAME@rd-mvb-linuxlab.bristol.ac.uk](mailto:USERNAME@rd-mvb-linuxlab.bristol.ac.uk) | 表示“跳过此主机”的-J

[uname -a] to print all information about the system

university username ([whoami] shows this) , [hostname] which just prints the machine name

**设置ssh keys，避免在以上步骤中输入两次密码**

two-factor authentication：使用密钥文件登录，该密钥文件本身受密码保护

convenient: you don't have to type your password anymore.

secure: key never leaves your machine, even if you connect to an evil server.

id\_CIPHER的文件中的**私钥**（也称为密钥），其中CIPHER是所使用的密码。

名为id\_CIPHER.pub的文件中的**公钥**，您需要将其副本存储在您想要登录的任何机器或任何服务上（对于实验室，因为实验室机器都共享一个文件系统，您只需要存储一次

生成密码对：[ssh-keygen -t ed25519]，通常保存在主目录中的.ssh文件夹；-t参数选择要使用的加密算法。

private key permission (-)(rw-)(---)(---)：除了您自己（和root）之外没有人可以读取您的密钥文件

public key permissions are (-)(rw-)(r--)(r--)：所有者可以读写，而组和其他所有人（假设他们有权访问文件夹）可以读取公钥

known\_hosts是SSH存储您已经连接到的计算机的公钥的地方：当您第一次连接到新计算机时，每次您对“您确定要连接吗？”问题回答“是”时，它都会将结果存储在这个文件中，并且下次不会再问您。 (-)(rw-)(r--)(r--)

安全复制到seis的 ~/.ssh目录下：scp~/.ssh/id\_ed25519.pub[USERNAME@seis.bris.ac.uk:~/.ssh/](mailto:USERNAME@seis.bris.ac.uk:~/.ssh/)；如果scp中包含冒号（:），那么它指的是另一台机器上的文件。

授权使用public key：

cat id\_ed25519.pub >> authorized\_keys 复制id\_ed25519到authorized\_keys文件

chmod 600 authorized\_keys；600表示rw-------

**为lab machine设置key**

agent forwarding：如果您SSH到一台机器，那么当您尝试进一步SSH到另一台机器时，SSH将重用相同的密钥。这样做的方法是使用**-A**命令行标志。

机器向您发送一个很长的随机数，SSH用您自己机器上的私钥对它进行数字签名，并将签名发送回来，远程机器可以用公钥验证签名。

scp ~/.ssh/id\_ed25519.pub "rd-mvb-linuxlab.bristol.ac.uk:~/.ssh/"

cat id\_ed25519.pub >> authorized\_keys

chmod 600 authorized\_keys

**减少输入用户名：**

SSH读取两个配置文件：一个在/etc/ssh/ssh\_config（/etc是POSIX程序通常存储全局设置的地方）为所有用户读取，一个在~/.ssh/config为每个用户读取。

站点https://www.ssh.com/ssh/config/或终端上的man ssh\_config | less包含文档（man表示手册页，less是一个程序，每次在页面上显示一个文件，并允许您滚动和搜索）。

在/.ssh中创建配置文件：

Host seis

HostName seis.bris.ac.uk

User USERNAME

Host lab

HostName rd-mvb-linuxlab.bristol.ac.uk

ProxyJump seis

User USERNAME

**Vagrant**

Vagrant是一个管理虚拟机的程序。使用一个名为Vagrantfile的配置文件，它可以下载和配置磁盘映像（它称之为box），并调用其他程序来运行它们。Vagrant不会自己运行虚拟机，因此您需要另一个程序，如virtualbox。

运行[vagrant up]命令。这将启动在当前文件夹中配置的虚拟机，如果它尚未下载，那么它也会为您下载。

运行[Vagrant ssh]以登录到您的虚拟机。如果它要求您输入密码，请使用vagrant。现在您应该看到虚拟机提示[vagrant@debian12:~$]。

要退出虚拟机，键入[exit]，这将使您回到主机上的shell。在主机上，[vagrant halt]干净地关闭虚拟机。

当您配置一台Vagrant机器时，这将在主机上创建一个密钥对，并将公钥加载到客户机中。私钥实际上在主机上的.vagrant/machines/default/virtualbox/private\_key文件中，公钥在客户机上的/home/vagrant/.ssh/authorized\_keys。所以[vagrant ssh]所做的就是启动ssh -i KEYFILE vagrant@localhost -p 2222.

你在实验机上的主目录下的文件存储在网络文件夹中，但如果许多学生在他们的主文件夹中创建了许多vm，这将占用大量空间，并且速度很慢。如果将虚拟机存储在每台机器的本地/tmp文件夹中，那么登录到另一台实验机器或重新启动，vm将消失。

对于您想要安装在实验机器中的VM上的任何软件，您应该将安装命令写入Vagrantfile中的配置脚本中，以便下次Vagrant必须设置VM时重新安装它。

其他文件需设置一个共享文件夹，您可以在VM上以/shared的方式访问它，它映射到主机上包含Vagrantfile的文件夹。

**Debian**

**The file system**

/bin代表二进制文件(binaries)，也就是你可以运行的程序。这里有很多命令，包括ls。

/bin仅用于启动系统所需的二进制文件—或者至少是需要在几个磁盘驱动器（如shell）中运行的最重要的二进制文件。

/usr/bin是大多数二进制文件所在的地方，它们是全局可用的，例如在一个组织的所有机器上都可用。

/usr/local/bin用于本地管理员安装的二进制文件，例如用于组织内的某个部门。

/usr及其子文件夹用于存放通常只读的数据，例如程序和配置文件，而不是临时数据或日志文件。它包含像/usr/bin或/usr/lib这样的子文件夹，这些子文件夹与根目录中的文件夹重复。Debian清理这个混乱的方法是使它的/bin只是一个到/usr/bin的链接.

绿色是可执行程序，蓝色是到另一个文件的链接。

[ls -l /bin] 每行的第一个字符表示文件类型，主要字符是-表示普通文件，d表示目录，l表示所谓的软链接。您可以在清单的末尾看到每个链接的链接位置。

/etc存储系统范围的配置文件，通常只有root（管理员帐户）可以在这里更改内容。例如，系统范围的SSH配置位于/etc/ SSH中。

/lib包含动态库- windows调用.dll文件，POSIX使用.so。例如“/lib/x86\_64-linux-gnu/lib .so.6”是C库，它允许C程序使用printf等函数。

/home为用户的主目录目录，例如默认用户“vagrant”为“/home/vagrant”。唯一的例外是root, /root。

/sbin（系统二进制文件）通常包含只有系统管理员才会使用的程序。例如，fdisk在磁盘上创建或删除分区，许多以fs命名的程序处理管理文件系统。/sbin/halt，以root（或其他您允许这样做的用户）身份运行，关闭系统；还有/sbin/reboot

/tmp是一个临时文件系统，它可以存储在RAM中而不是磁盘上（但在必要时可以交换出来），并且它不需要在重新启动机器后存活下来。

/var保存随时间变化的文件，如日志或缓存。

/dev、/sys和/proc为虚拟文件系统。UNIX设计原则之一是，几乎所有与操作系统的交互都应该看起来像读写文件一样，或者简而言之，*一切都是文件*。例如，/dev为诸如硬盘（/dev/sda是系统中的第一个SCSI磁盘，/dev/sda1是该磁盘上的第一个分区）、内存（/dev/mem）和一些伪终端或ttys等设备(devices)提供接口。/proc提供对运行进程(process)的访问；/sys提供对系统功能的访问。例如，在某些笔记本电脑系统上，写入[/sys/class/backlight/acpi\_video0/brightness]会改变屏幕亮度。

/shared文件夹不是FHS的一部分，但它是这个单元与Vagrant虚拟机上的主机共享文件夹的约定。在前几年，我们称这个文件夹为/vagrant，但鉴于虚拟机的默认用户名也是“vagrant”，这导致了很多混乱，所以我们改变了它。

**Package managers**

存储库(repository)是你可以安装的软件的集合，它可以被托管在任何地方。包管理器是从存储库中安装包的软件，主要作用是包可以依赖于其他包，当您安装一个包时，它会自动安装依赖项。

sudo apt install emacs-nox

Sudo（superuser do）允许您以root身份运行命令。如果您真的需要一个root shell，那么可以使用sudo bash，使用#而不是$提示符来警告您正在以root身份工作。

apt是Debian包管理器。

nano, vim, emacs, mcedit, micro

可以使用apt info PACKAGE查找有关包的信息

使用sudo apt remove PACKAGE将它们从系统中删除。

Sudo apt update从存储库中获取新的包列表, 自上次检查以来是否有任何软件包已更新到新版本。

Sudo apt upgrade将您已经安装的每个包升级到本地包列表中的最新版本（在执行apt更新时下载）。

在虚拟机上应该把软件放在Vagrantfile中，然后vagrant up会自动安装。您的Vagrantfile已经包含了一个以echo开头的行，您可以在这个行下面放一个apt-get install PACKAGE行。这里没有sudo，因为当Vagrant安装系统时，它会自动以root身份运行。

1. **Fundamentals**

**Shell**

Terms：shell, terminal, console, command line, (command) prompt, xterm, rxvt, konsole, (gnome)-terminal, putty (Windows)

workflow :The user creates a request in the form of a typed command, and the shell processes this and produces a response.

**$** You are in a shell, most likely POSIX(sh) compatible.

**#** You are in a root shell. With great power comes great responsibility.

**%** You are probably in the C shell.

**>** You are on a continuation line e.g. inside a string.

**TAB**: complete command or filename

**DOUBLE** **TAB**: show list of possible completions

**UP/DOWN**: scroll through history

**^R text**: search history for command

**which** ls: /bin/ls 在环境变量$PATH设置的目录里查找符合条件的文件

**ls**: file1 file2；–l: -rwx------ 1 vagrant ... 40 ... file1以长格式显示文件和目录信息，包括权限、所有者、大小、创建时间等。–a: . .. file file2显示所有文件及目录 (**.** 开头的隐藏文件也会列出)；–help

**man** [SECTION] COMMAND Section 1 is shell commands, section 2 system calls, section 3 the C library etc.

On lab machines: fairly user-friendly manual. On alpine: programmer's manual.

shell deals with expanding pattern，program deals with its arguments

**\*** all filenames. a\* is filenames starting with a

**?** single character. image???.jpg matches image001.jpg

**[ab]** single character in list. image[0-9].jpg

**$** variable name expansion：${a}b means the value of the variable a followed by the letter b, whereas $ab would mean the value of the variable ab

**双引号**里可以有变量，双引号里可以出现转义字符 turn off pattern matching

**单引号**里的任何字符都会原样输出，单引号字符串中的变量是无效的；单引号字符串中不能出现单独一个的单引号（对单引号使用转义符后也不行），但可成对出现，作为字符串拼接使用。

**cp [-rfi] SRC... DEST** copy files**-r** recursive**-f** overwrite readonly**-i** ask before overwriting (interactive) source为源文件或路径，可以有很多个，最后一个默认为dest。

cp index.html style.css web；cp \* web

*in empty folder:* $ cp \* web；cp: can't stat '\*': No such file or directory

**mv [–nf] SRC... DEST**move files**-n** no overwrite**-f** force overwrite

**find DIR [EXPRESSION]：**find all files in directory (recursively)that match an expression

find . –name "a\*"

**lab**

./arguments one two ./arguments "one two" ./arguments "one two"

1.Argument #0: [./arguments] Argument #1: [one] Argument #2: [two]

2. Argument #0: [./arguments] Argument #1: [one two]

3. Argument #0: [./arguments] Argument #1: [one two]

Argument #0: [./arguments] Argument #1: [\*]

1. ./arguments "\*" 2. ./arguments \\* 3. ./arguments '\*'

touch silly named file The command **without quotes** creates three separate files

program="silly name"

gcc -Wall $program.c -o $program -> gcc -Wall silly name.c -o silly name

program="silly name"

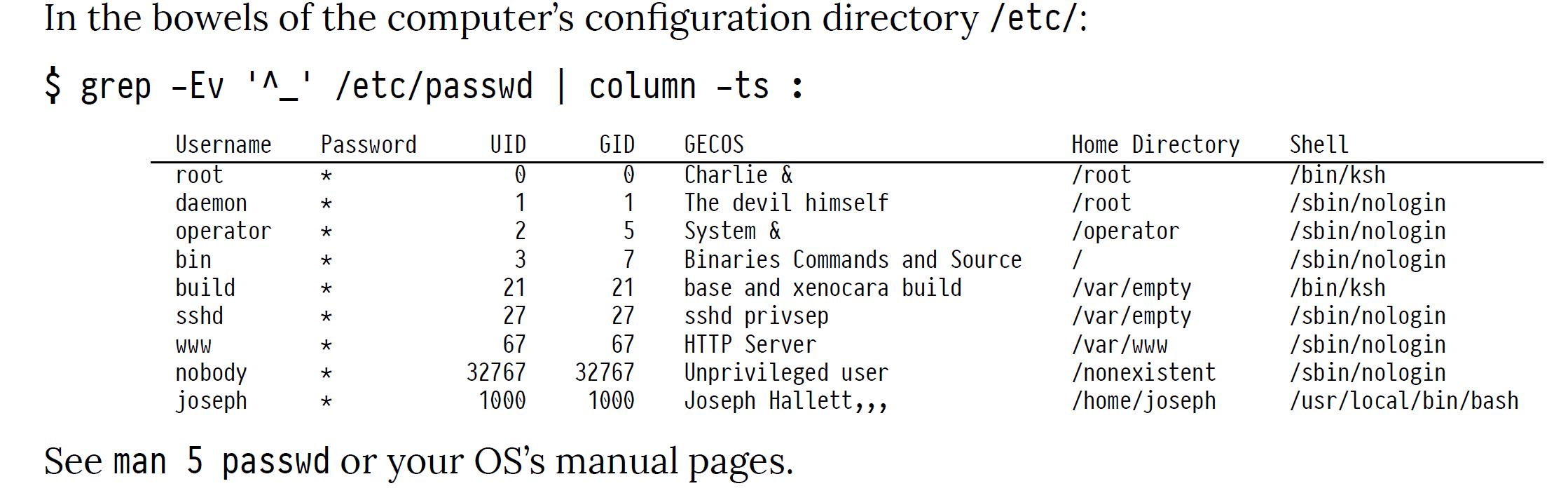
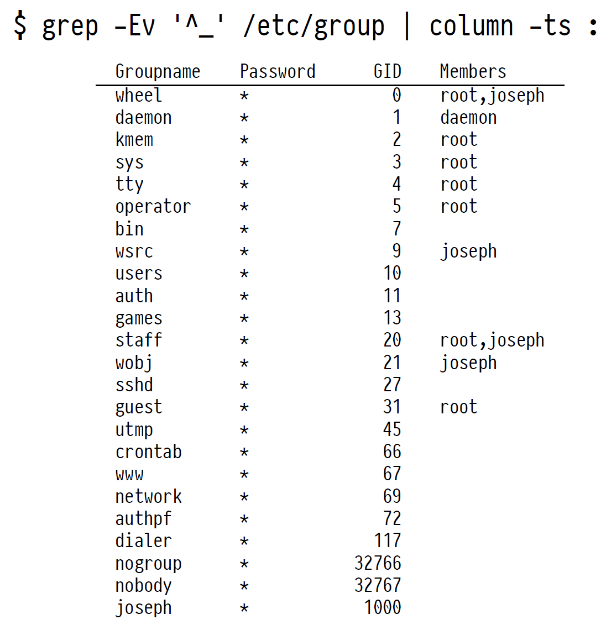
gcc -Wall "$program.c" -o "$program" -> gcc -Wall "silly name.c" -o "silly name"

program=silly name -> set program = silly, then execute the program name.

file=arguments gcc -Wall "$file.c" -o "$file" -> need to set the variable on a separate line.

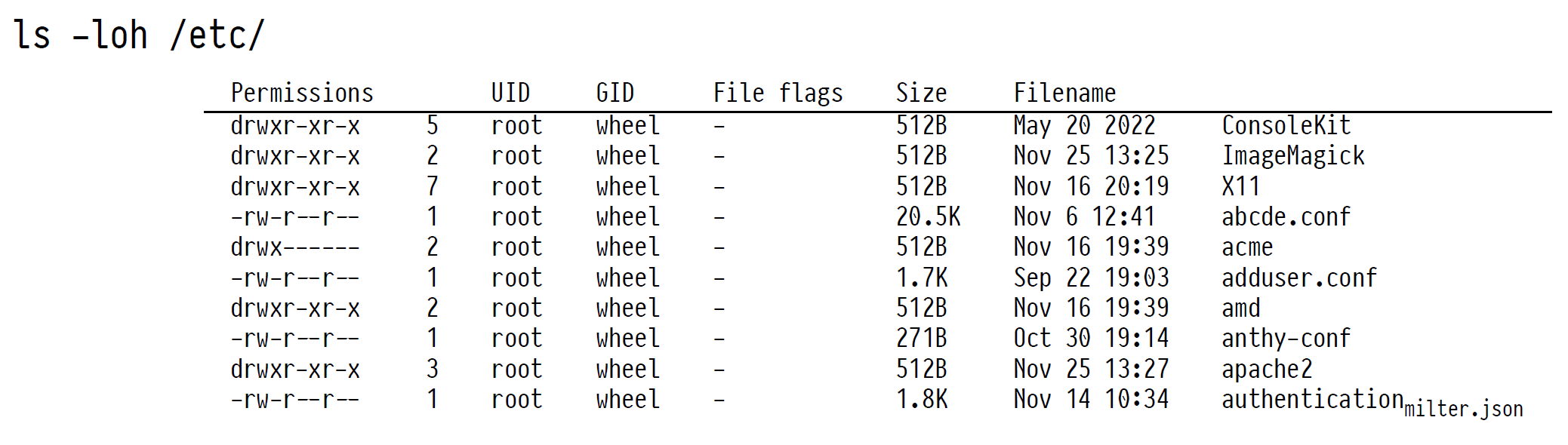
**Permissions**

Root：The *super user*，The *system administor*，UID 0，root beget init which beget login

  inside the password file

inside the group file

Each group can have *multiple* members；No passwords ever actually listed；They’rein /etc/shadow

 all files were owned by a user and a group

File type

- regular file；b special file；c special file；d directory；l symbolic link；p FIFO；s socket link

Read bits：r can read；- cannot read

Write bits：w can write；- cannot write

Execute bits

x can execute (or cd into for directories)；s run as UID/GID of owner（user/group）; t sticky bit (append only directory)(other)；- cannot execute

The *sticky* bit t is mostly for log directories and temporary directories. You should be able to append to log files, but not delete them.

The setuid/setgid bits are used for privilege separation.

ls -l/etc/spwd.db；-rw-r—1 root\_shadow40960Dec2215:03/etc/spwd.db

The passwd program changes your password: ls -l$(command-v passwd)：

-r-sr-xr-x 1 root bin 21208 Jan 12 03:08 /usr/bin/passwd

su switch to user (by default root) with *their* password

sudo switch to user *if the sysadmin says you’re allowed to* with *your* password

doas modern rewrite of sudo with less bugs and Spiderman references

**change who owns a file:**

chown [-cfhvR] [--help] [--version] user[:group] file...

把 /var/run/httpd.pid 的所有者设置 root：

chown root /var/run/httpd.pid

将文件 file1.txt 的拥有者设为 runoob，群体的使用者 runoobgroup :

chown runoob:runoobgroup file1.txt或chown :runoobgroup file1.txt

chown joseph:staff exam # Alternatively... chown :staff exam

**change a file’s permissions：**

chmod go-wx exam

u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。

+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。

r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。s setuid/gid

7：rwx；6：rw-；5：r-x；4：r--；3：-wx；2：-w-；1：--x；0：---

传统上，根用户可以做任何事情。在大多数现代操作系统中，这种情况已经发生了更大的变化。例如，Linux使用功能来设置任何用户可以做的事情，并使用命名空间来允许具有不同功能的多个根用户

**Lab**

[sudo adduser NAME]创建一个新用户

- **tail**默认显示文件的最后10行；tail -n N FILE将显示最后N行。用tail /etc/passwd和tail /etc/group检查用户和组文件

su USERNAME 命令用于变更为其他使用者的身份，除 root 外，需要键入该使用者的密码。

sudo usermod -aG GROUPNAME USERNAME -G<群组> 　修改用户所属的附加群组

chgrp -R GROUPNAME DIRECTORY 将目录的组更改为users。**-R 或 --recursive**： 　递归处理，将指定目录下的所有文件及子目录一并处理。

非根挂载和卸载规则

根据手册页面，如果满足特定条件，则允许非root用户执行某些类型的挂载和卸载：

非根安装

用户选项

如果/etc/fstab中的文件系统条目指定了用户选项，则非root用户可以挂载文件系统。

挂载的文件系统不能覆盖敏感的系统路径（例如，/,/home）。

非根卸载

用户选项

非root用户挂载的文件系统也可以被该用户卸载，只要在/etc/fstab中指定了user选项。

这确保用户不能卸载关键系统文件系统或其他用户挂载的文件系统。

限制:

用户不能卸载他们自己没有挂载的文件系统，除非在非常特定的配置中，比如/etc/fstab中的用户标记挂载。

为什么允许非root挂载/卸载？

方便：允许用户挂载个人USB驱动器或网络共享，无需管理员干预。

受控访问：通过限制可以使用哪些文件系统和挂载点，可以将安全风险降到最低。

委派：允许受信任的用户管理特定的资源，而无需授予完全的根权限。

sudo cat /etc/sudoers

root ALL=(ALL) ALL

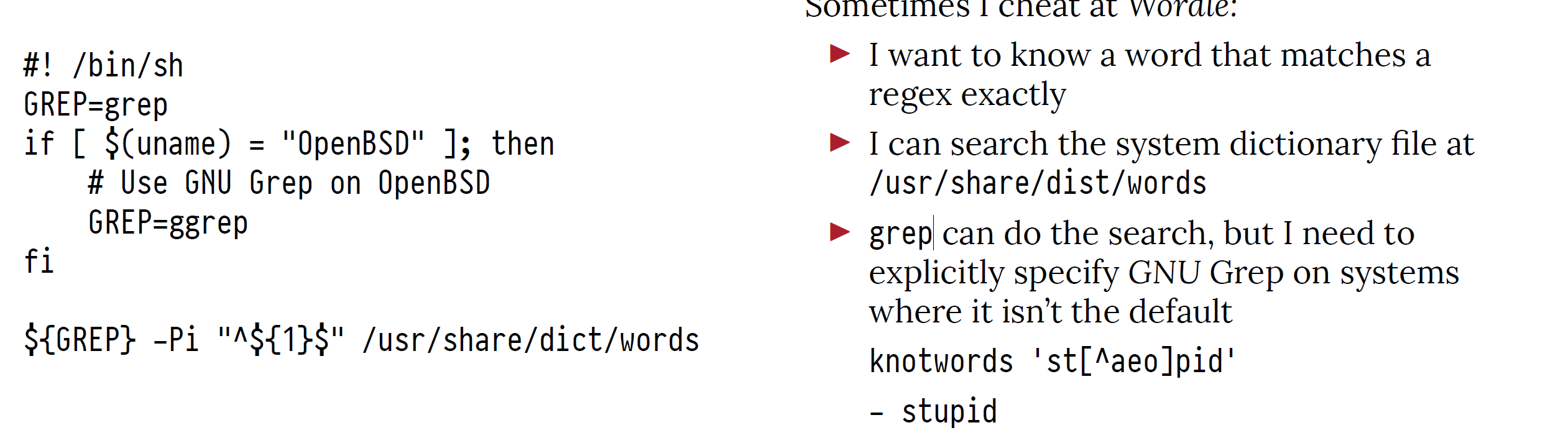
#includedir /etc/sudoers.d，包含一个单独的文件vagrant：vagrant ALL=(ALL) NOPASSWD: ALL，这就是为什么vagrant可以首先使用sudo。

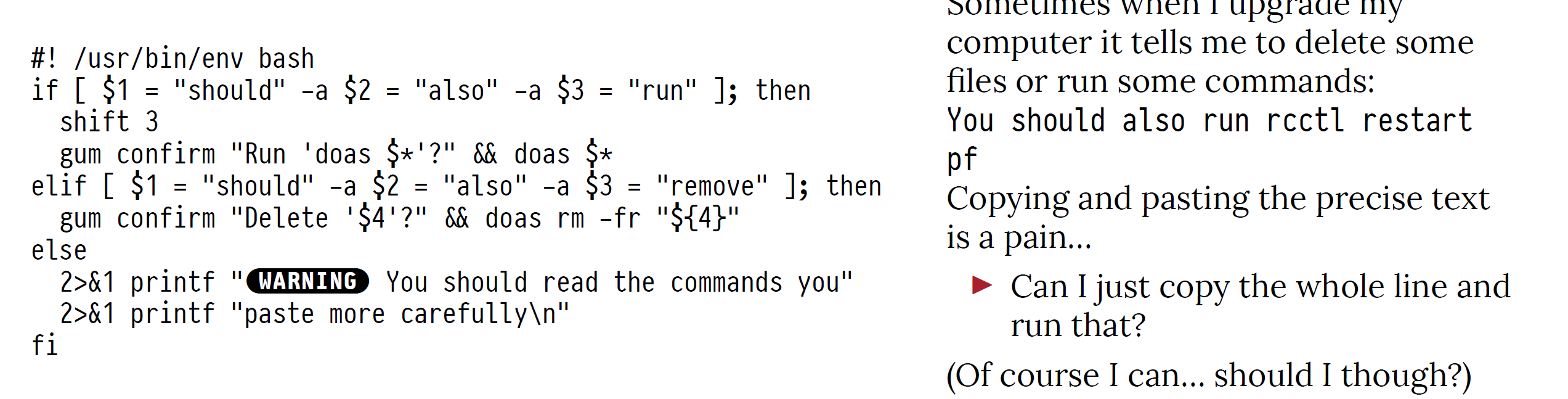
以visudo作为root用户编辑sudoers文件，并添加以下行：

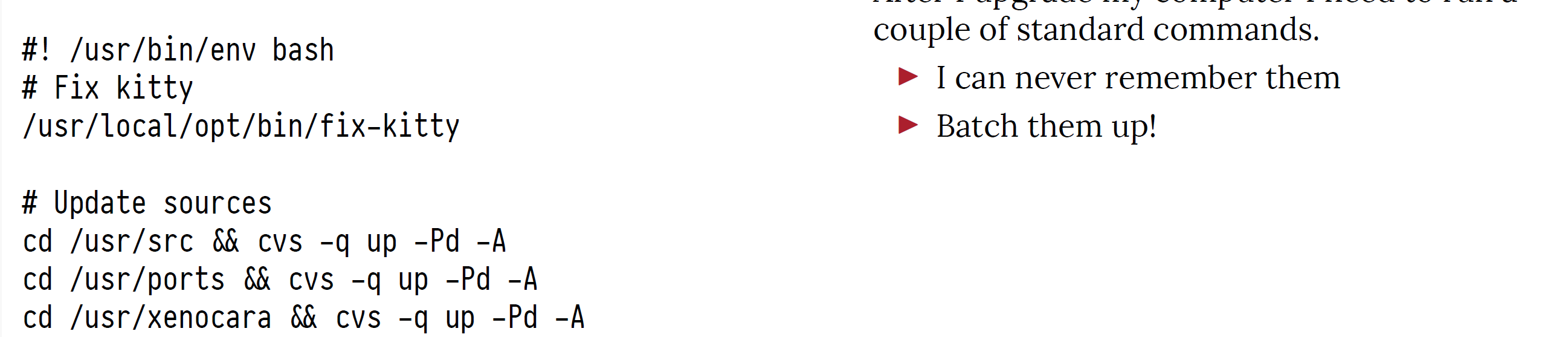
%users ALL=(ALL) /sbin/reboot，users组中的每个人使用sudo运行命令

**Shell script**

Saves a tonne of time；Often easier than writing a full program







*Shebang：* #!

path to the interpreter of the script plus any arguments：/bin/sh/或/usr/bin/env bash

env [-i] [name=value ...] [utility [argument ... ]]

Env在修改命令行上指定的环境后执行实用程序。选项name=value指定一个环境变量name，其值为value。env所做的是遍历PATH并尝试找到指定的程序并运行它。

Colon separated list of paths



chmod +x my-script.sh

./my-script.sh or sh my-script.sh

https://www.shellcheck.net

A; B run A then run B

A | B run A and feed its output as the input to B

A&&B run A and if successful run B

A || B run A and if not successful run B

Programs *return* a 1 byte exit value (e.g. C main ends with return 0;) This gets stored into the variable ${?} after every command runs. 0 indicates success (usually)；>0 indicates failure (usually)：

do\_long\_running\_command

test $? -eq 0 && printf "Command succeeded\n"或[ $? -eq 0 ] && printf "Command succeeded\n"

[ initiates the test，] marks the end of the test。There is **a required space** after the [ and before the ] because [ is treated as a standalone command.

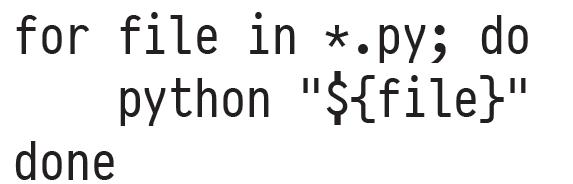
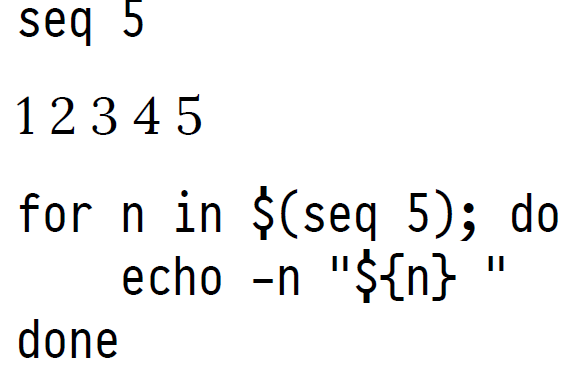
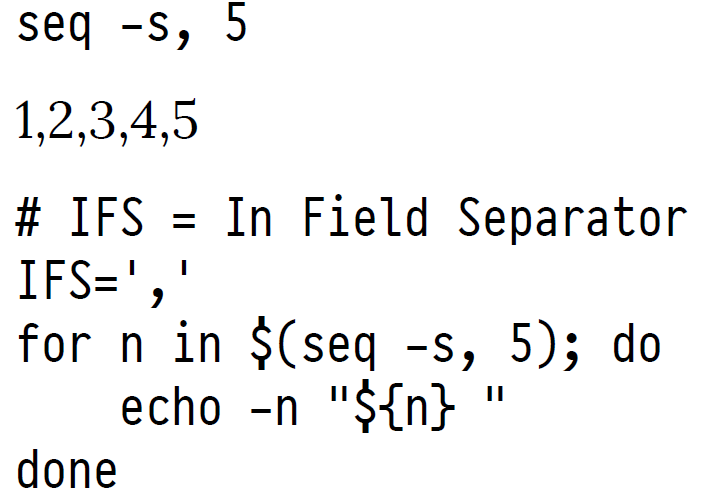
Different shells:

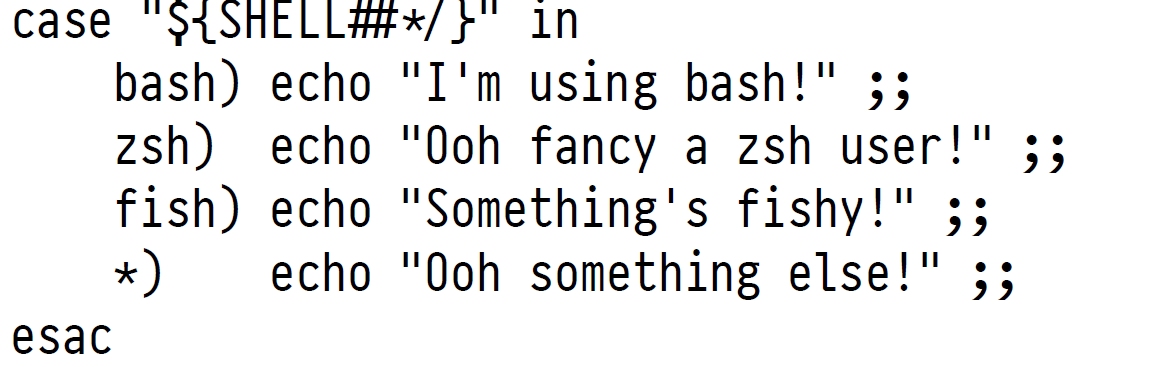
sh POSIX；bash Bourne Again shell；zsh；ksh；dash；Busybox sh；fish；elvish；nushell

**pipe**

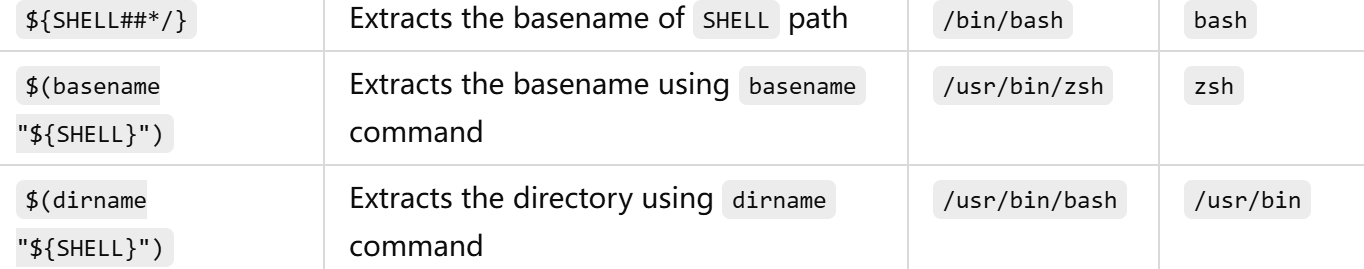
GREETING="Hello World!" (No spaces around the =)；echo "${GREETING}"；*environment variable*: export GREETING；get rid of a variable：unset GREETING；没有定义变量，引用时会引用为‘’

${0} Name of the script；${1}, ${2}, ${3}… Arguments passed to your script；${#} The number of arguments passed to your script；${@} and ${\*} All the arguments



echo "${SHELL}"；echo “${basename%.\*}”获取 . 前的文件名



for f in \*.jpg; do

convert "${f}" "$(basename "${f}" .jpg).png"

done

**ps** -A | grep -i firefox | awk '{print $1, $5}' | ghead -n -1 | wc -l

ps （process status）命令用于显示当前进程的状态，-A 列出所有的进程

**grep** (global regular expression)查找正则表达式。-i：忽略大小写进行匹配。

**awk** 是一种处理文本文件的语言，是一个强大的文本分析工具。可以指定需要哪些行

**wc**命令用于计算字数。-l或--lines 显示行数。

如果您只想知道特定命令或管道产生多少结果（假设每行产生一个结果），那么wc - 1。

The **|** pipe copies *standard output* to *standard* ***input***…

The **>** pipe copies *standard output* to a named **file**…

The **>>** pipe **appends** *standard output* to a *named* ***file***…

The **<** pipe **reads** a **file** *into* standard **input**… (e.g. grep firefox <processes.txt)

The **<<<** pipe takes a string and *places it on standard* ***input***

**2>&1** to a command will run it with ***standard error* merged into *standard output***)

**lab**

set -e makes the whole script exit if any command fails. It is like putting || exit $? on the end of every command.

set -u表示引用未定义的变量是错误的。

set -o pipefail改变管道的工作方式：通常，管道的返回值是管道中最后一个命令的返回值。使用pipefail选项，如果管道中的任何命令失败（非零返回），则管道返回该命令的退出代码。

**Pipe**

Unix Philosophy：It is easier to maintain 10 small programs than one large program. Therefore, Each program should do one thing well. Programs should be able to cooperate to perform larger tasks. The universal interface between programs should be a text stream.

#define STDIN\_FILENO 0；#define STDOUT\_FILENO 1； #define STDERR\_FILENO 2

**grep** [options] pattern [files] -i：忽略大小写进行匹配；-n：显示匹配行的行号；-H 或 --with-filename；-v：反向查找，只打印不匹配的行；-w 或 --word-regexp : 只显示全字符合的列；-o 或 --only-matching : 只显示匹配PATTERN 部分；-c：只打印匹配的行数。

**sort** [-bcdfimMnr][-o<输出文件>][-t<分隔字符>][+<起始栏位>-<结束栏位>][--help][--verison][文件][-k field1[,field2]]

-r 以相反的顺序来排序。

**cat** infile | sort > outfile 等于 **sort** < infile > outfile

**error redirect**

$ COMMAND > FILE 2> FILE2或$ COMMAND > FILE 2>&1

**ignore output:**

$ COMMAND > /dev/null

A program that uses a standard stream can be told to use a file instead by：

1. PROGRAM < FILE (standard input)；PROGRAM > FILE(standard output)；
2. PROGRAM 2> FILE(standard error)

A program that expects a filename can be told to use standard input/output instead by:

1. using the filename **–** (single dash), if the program supports it
2. using the filename /dev/stdin etc., if your OS supports it

cat ./-；rm ./-f

**tee** takes a filename as argument and writes a copy of input to it

tee [-ai][--help][--version][文件...]

**less** ：Up/Down arrows scroll；Space/Enter advance a page；/ (forward slash) opens a search,q quits；less [参数] 文件

If PROGRAM wants a file to read from, how can I pipe something in：

**$ cat <(echo "Hi")**

Hi

$ echo <(echo "Hi")

/dev/fd/63

echo $(echo Hi | sed –e s/Hi/Hello/)

Hello

**Lab**

**cat**（concatenate）命令用于连接文件并打印到标准输出设备上

cat [FILENAME [FILENAME...]] writes the contents of one or more files to standard output. 如果去掉所有文件名，cat将读取其标准输入并将其写入标准输出。

**head** 命令可用于查看文件的开头部分的内容，有一个常用的参数 **-n** 用于显示行数，默认为 10。head -n -2 to *skip the last 2 lines* and write all the rest.

**uniq** 命令用于检查及删除文本文件中重复出现的行列

**diff** 命令用于比较文件的差异。

**diff** [-abBcdefHilnNpPqrstTuvwy][-<行数>][-C <行数>][-D <巨集名称>][-I <字符或字符串>][-S <文件>][-W <宽度>][-x <文件或目录>][-X <文件>][--help][--left-column][--suppress-common-line][文件或目录1][文件或目录2]

1. **Regex**

^X只匹配以X开头的字符串

‘X$’ 匹配以X结尾的字符串

. 匹配一个字符

‘.ench.’匹配一个及以上字符+ench+一个及以上字符

‘^..ench..$’匹配两个字符+ench+两个字符

\转义符，‘\^’包含^的字符串

[^AEIOUaeiou]匹配不含AEIOUaeiou，[]表示或者

‘a\*’将查找具有任意数量的字母a的单词，包括0

'(way|day)$'，| 表示或

encyclo(p|ae)dia

colou?r

Sed DEST模式可以使用字符&引用复制SOURCE中的匹配内容。

sed 's/\(is \)\+/at was /' 加号表示重复一次或多次前面的模式。‘is ‘->’at ‘’was ’

sed 's/^t\(.\*\)/\1s/将从任何以t开头的行前面去掉’t‘，并在末尾加上’s'。

sed 's/\(.\*\)\(way|day\)/\2 (\1)/' 1表示前面括号里的.\*，2表示后面括号里的way|day

sed 's/\([^ ]\+\) \(.\*\)/\2 "\1"\?/' [^ ]表示不是空格，.\*剩下所有的

This is the best -> is the best “This”?

s/a/e/和s/a/e/g的区别 g表示替换所有a为e

正则表达式遵循构成*形式语言*的规则。*与自然语言*不同，形式语言完全由描述其如何运作的规则所封装。计算机科学的基本原则：定义一组正式的规则 *等同*于描述一台机器。语法是一种判断（解析）字符串是否有效的机器。

*正则表达式定义了一种来自称为正则语言*的语言类的语法 。语言类型有如下层次：

正则 < 上下文无关 < 上下文相关 < 递归可枚举

所有正则语言都是上下文无关的，但并非所有上下文无关语言都是正则的（等等）。

对于一种可递归枚举的语言（但不是更简单的语言），唯一能够解析它的机器是**图灵机**——一种在（理论上）无限磁带上运行的功能齐全的“计算机”。**上下文相关语言可以通过在稍微更合理的限制（“线性有界自动机**”）下运行的图灵机形式来解析。上下文无关语言可以通过**下推自动机**（一种可以访问*堆栈*的简单机器）进行解析。*正则语言*可以用**有限状态自动机**来解析。

简单来说：您可以列出这台机器将处于的所有状态，以及所有将导致状态转换的输入。

实际上可以将 FSM/FSA 表示为状态和转换的简单表。

有限状态机是*非常有限的*机器。有些东西**无法**用状态转换图来表达。

一个简单的例子：匹配任意数量的字符<>，能表达已知量： <(things)>、<<(things)>>

可以表达 <\*(things)>\*——任意数量的<，以及任意数量的>，但是我们不能让第一个和第二个数量相等，因为 FSM*没有内存*。您需要一个*堆栈*（或等效物）来跟踪这样的数字。

In practice, though, HTML opening tags seen and handled in the wild are often not well-formed, and there are a lot of difficult cases to consider.

1. **Git**

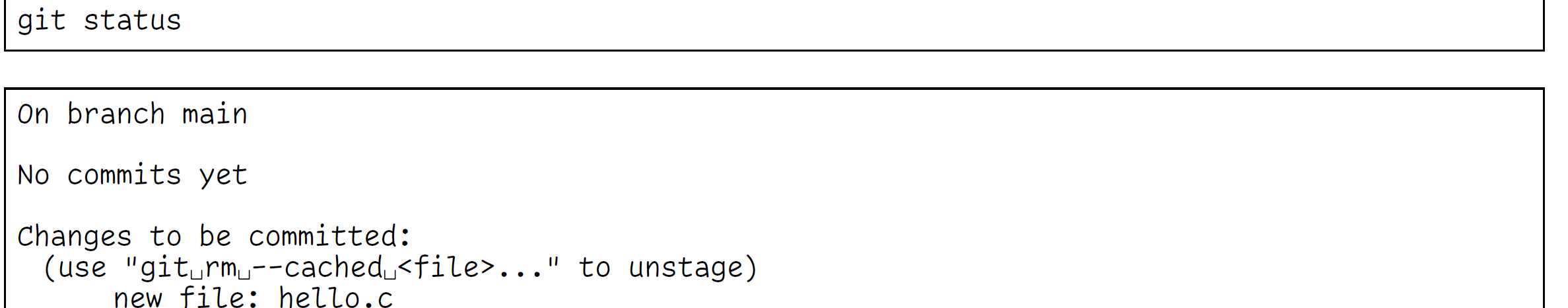
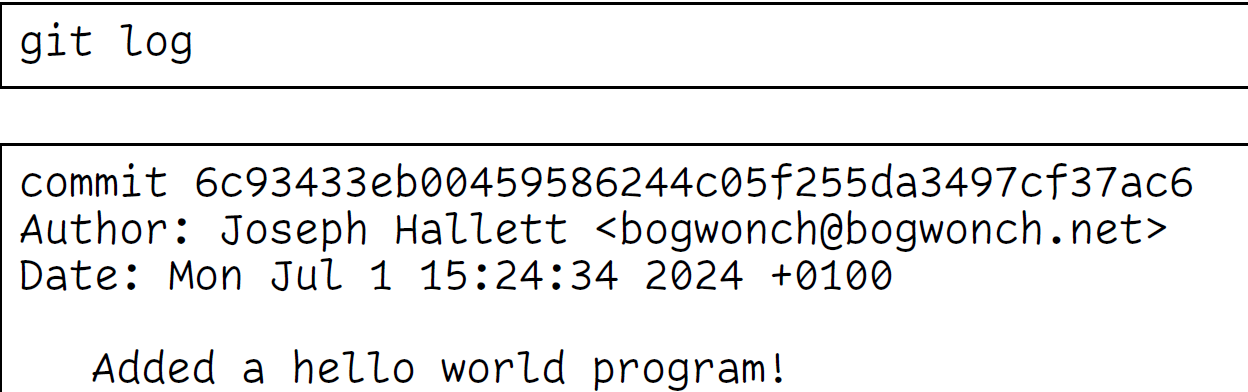
git **init** - 初始化仓库。

git **add** - 添加文件到暂存区。

git **commit** - 将暂存区内容添加到仓库中。

git **clone** - 拷贝一份远程仓库，也就是下载一个项目。

git **status** 查看仓库当前的状态，显示有变更的文件。

git **log** 查看历史提交记录 -–remotes --oneline

查看已缓存的与未缓存的所有改动：git **diff** HEAD

git **remote** -v：列出当前仓库中已配置的远程仓库，并显示它们的 URL。

git remote add <remote\_name> <remote\_url>：添加一个新的远程仓库。指定一个远程仓库的名称和 URL，将其添加到当前仓库中。

git **push** <远程主机名> <本地分支名>:<远程分支名>。上传远程代码并合并

如果本地分支名与远程分支名相同，则可以省略冒号：

git push <远程主机名> <本地分支名>

Where to get help (apropos git) **apropos** git -a tutorial

**git checkout <branch-name>或hash的前6个字符** 命令用于在不同的分支之间切换、恢复文件、创建新分支等操作。-b创建新分支并切换

**git switch <branch-name>或hash的前6个字符**

切换分支/版本后，可以查看当前版本的文件信息

git **blame** <file> - 以列表形式查看指定文件的历史修改记录。包括修改hash，用户，时间，内容

$ git **reset** HEAD^ # 回退所有内容到上一个版本

$ git reset HEAD^ hello.php # 回退 hello.php 文件的版本到上一个版本

$ git reset 052e # 回退到指定版本

git **revert** HASH添加一个新的提交，使用给定的哈希值将文件返回到提交之前的状态。这在团队开发期间是安全的，因为它只是添加一个新的提交。如果你有提交A， B，并执行git revert B，那么你会得到一个新的提交C，所以任何使用存储库的人都可以看到提交A， B， C的序列；但是C中的文件状态和A中的是一样的

**git fetch** 命令用于从远程获取代码库。to update the remote branches：git fetch --all

将其他分支合并到当前分支：git **merge** <branchname>

**git pull** 其实就是 **git fetch** 和 **git merge** 的简写，先从远程仓库获取最新的提交记录，然后将这些提交记录合并到你当前的分支中。

**git rebase** 命令用于将一个分支上的更改移到另一个分支之上。它可以帮助保持提交历史的线性，减少合并时的冲突。git rebase <branchname>。

git rebase -i <commit>

pick：保留提交；reword：修改提交信息；edit：编辑提交；squash：将当前提交与前一个提交合并；fixup：将当前提交与前一个提交合并，不保留提交信息；drop：删除提交

git **format-patch** origin/master：将多个提交压缩成一个补丁文件。git **format-patch HEAD ^**代表最近一个文件的patch，有几个^则几个文件。Patch文件内容即diff

git **apply**：Applying patch files。跟踪传递patch，不需要fetch和push

git-**am** 可以一次合并一个文件，或者一个目录下所有的patch，或者你的邮箱目录下的patch.

git **config** core.excludesFile：Setting a .gitignore per repo

git **branch** -a, the -a (all) option means include branches that only exist on the remote

如果你达到一个重要的阶段，并希望永远记住提交的快照，你可以使用 **git tag** 给它打上标签。 git tag version-1.0

不同的人编辑了不同的文件git **merge** <branchname>

处理相同的文件

git **bisect**用来查找哪一次代码提交引入了错误。git bisect start [终点] [起点]，代码库就会切换到这段范围正当中的那一次提交

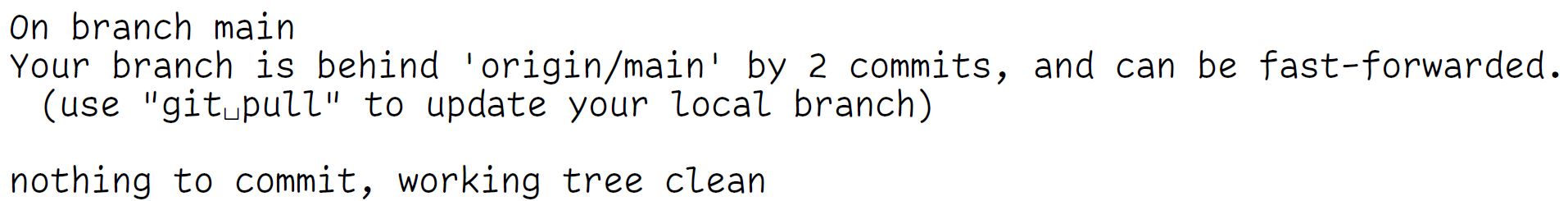
Where to get help(**apropos** git)

A Git ***forge***is aplacewhereyoucanhostaremote

HEAD -> main表示HEAD指针正在跟踪main

main and origin/main are *not the same branch，But they may have a similar history*

origin/main is telling you that Git is aware that at the remote origin there is also a branch called main. Last time git fetch-ed from that remote, the main branch at origin was pointing there。

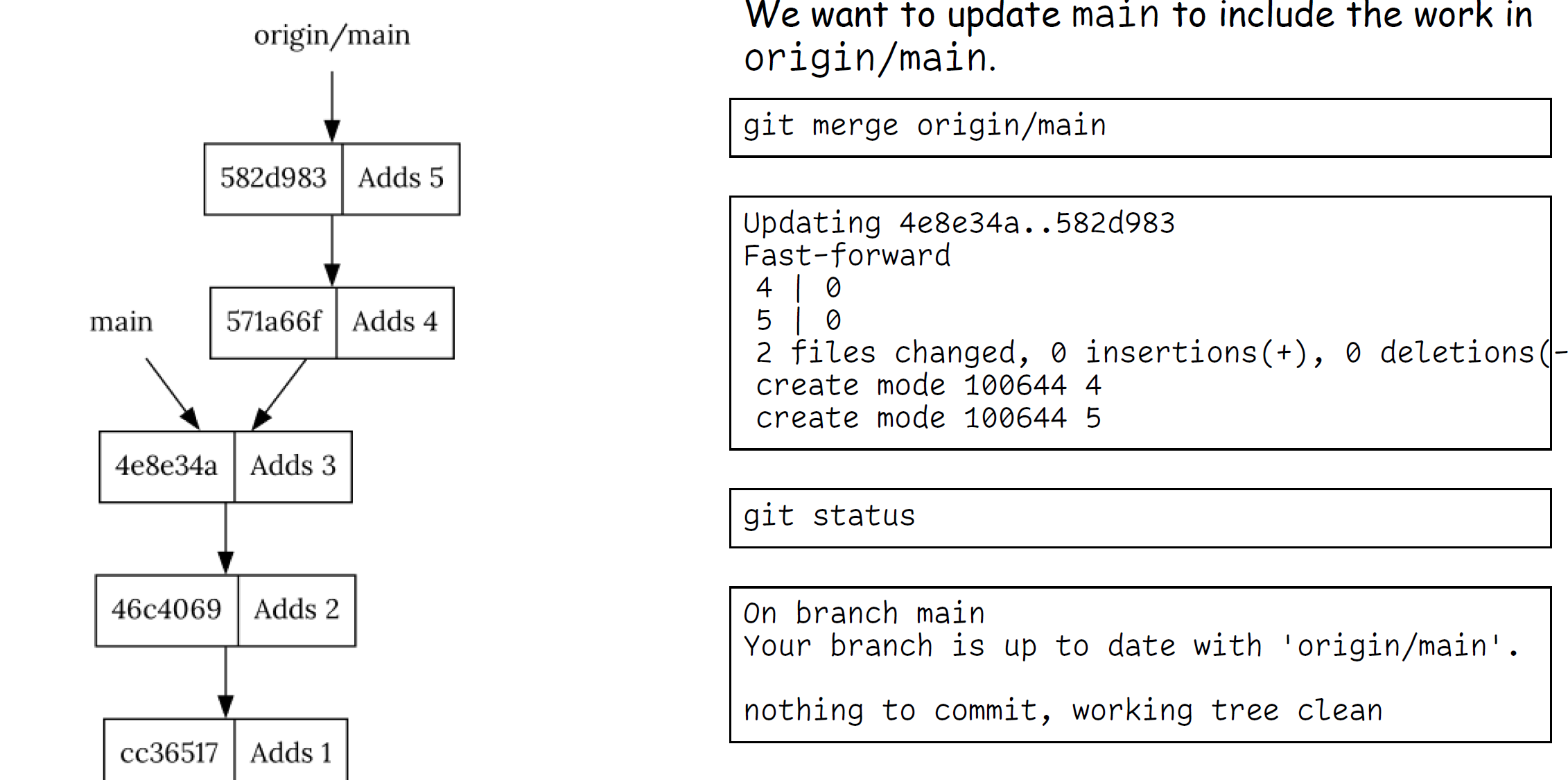
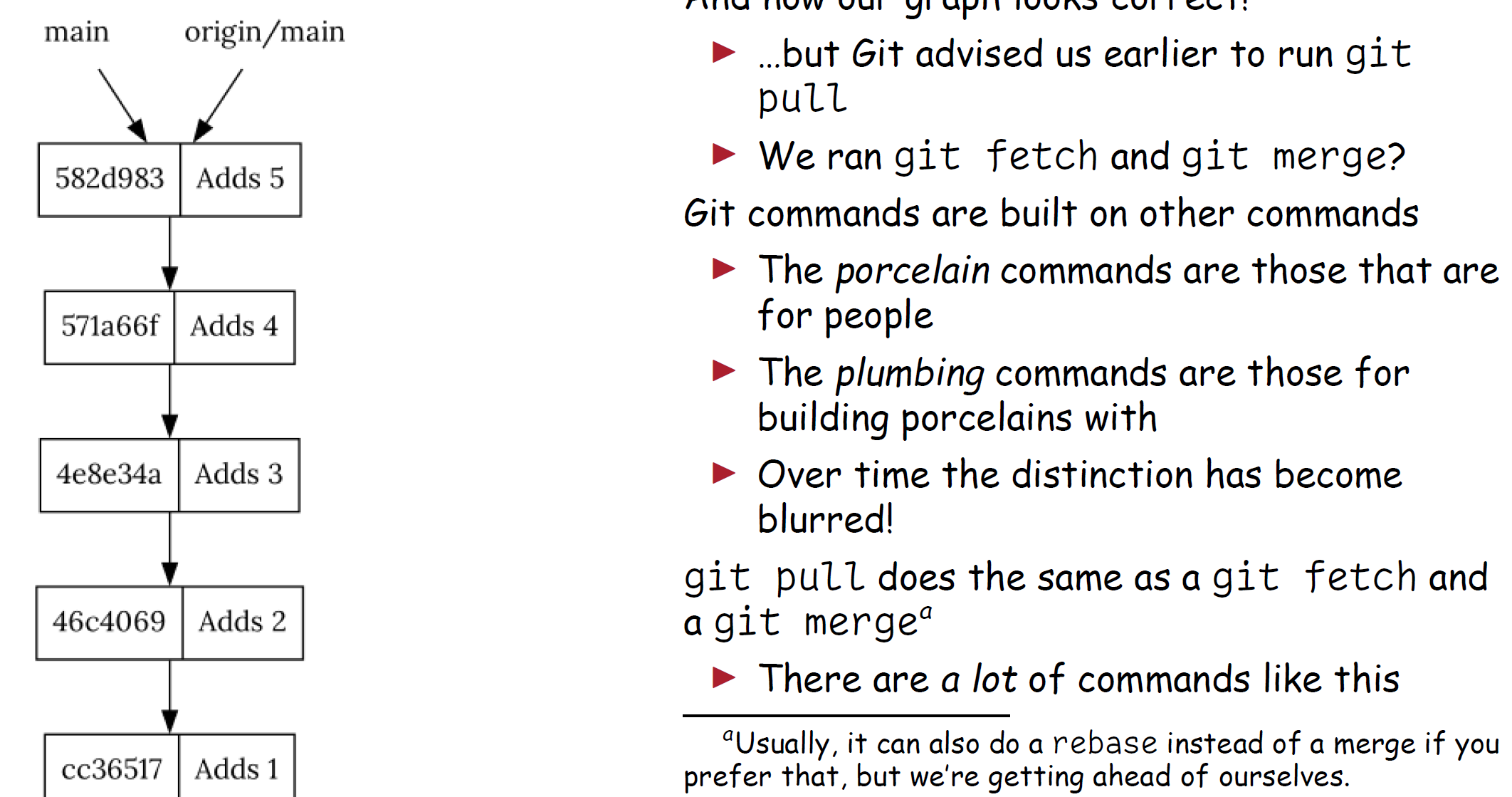


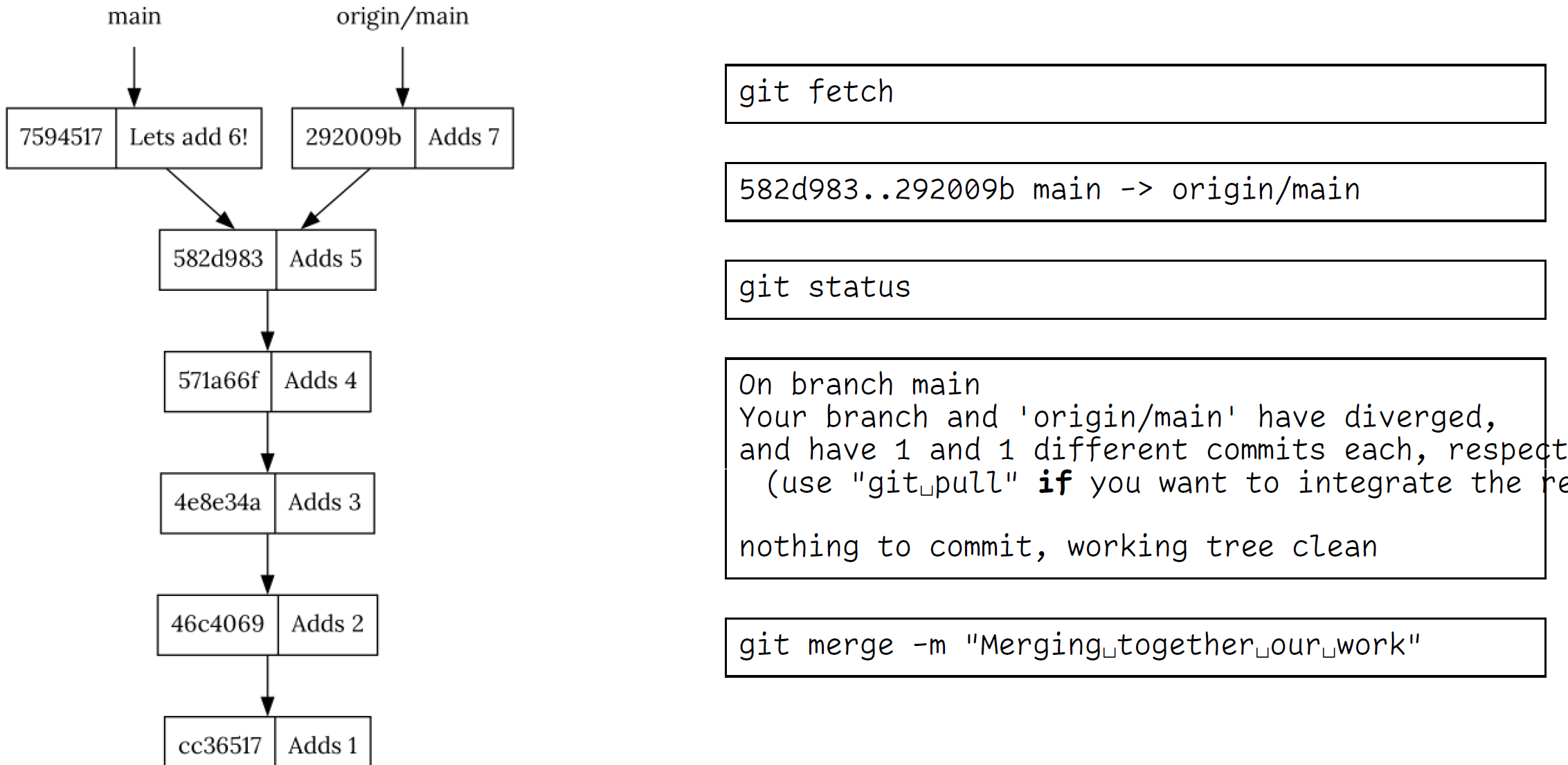
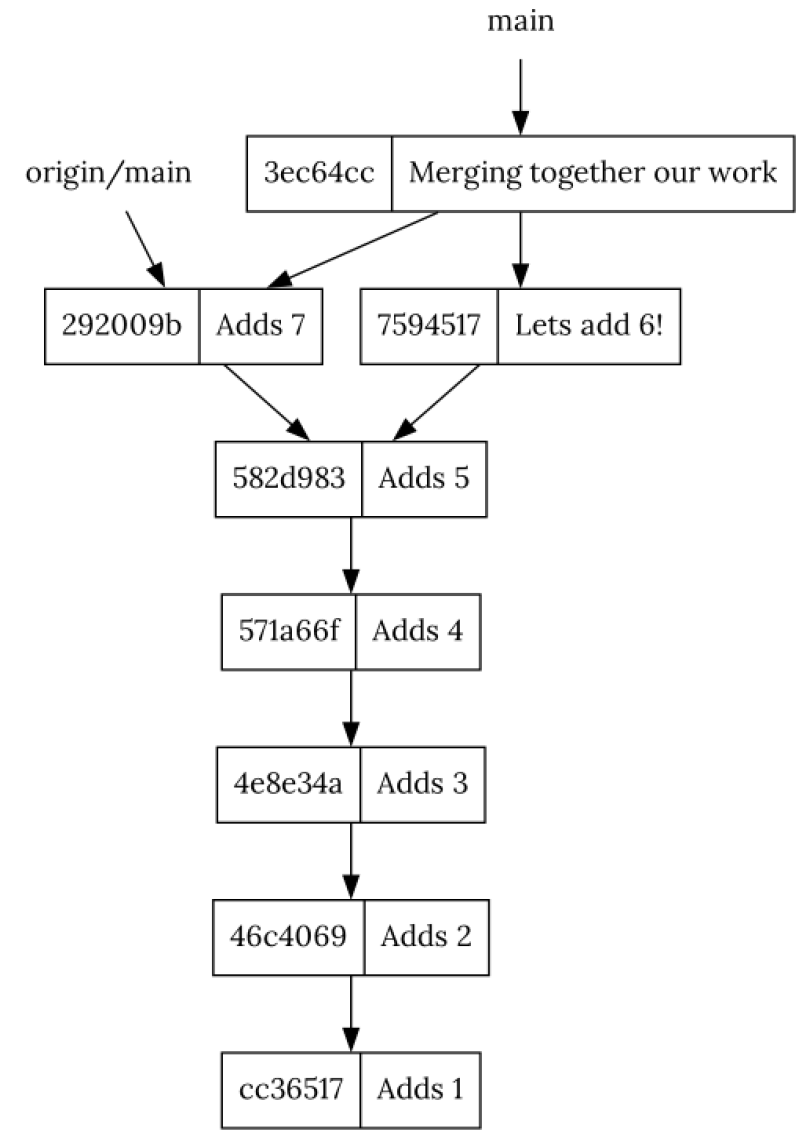
The team member who created the develop branch should now make a commit on it.

This branch currently exists only in their local repository, and if they try and push they would get a warning about this. What they need to do is

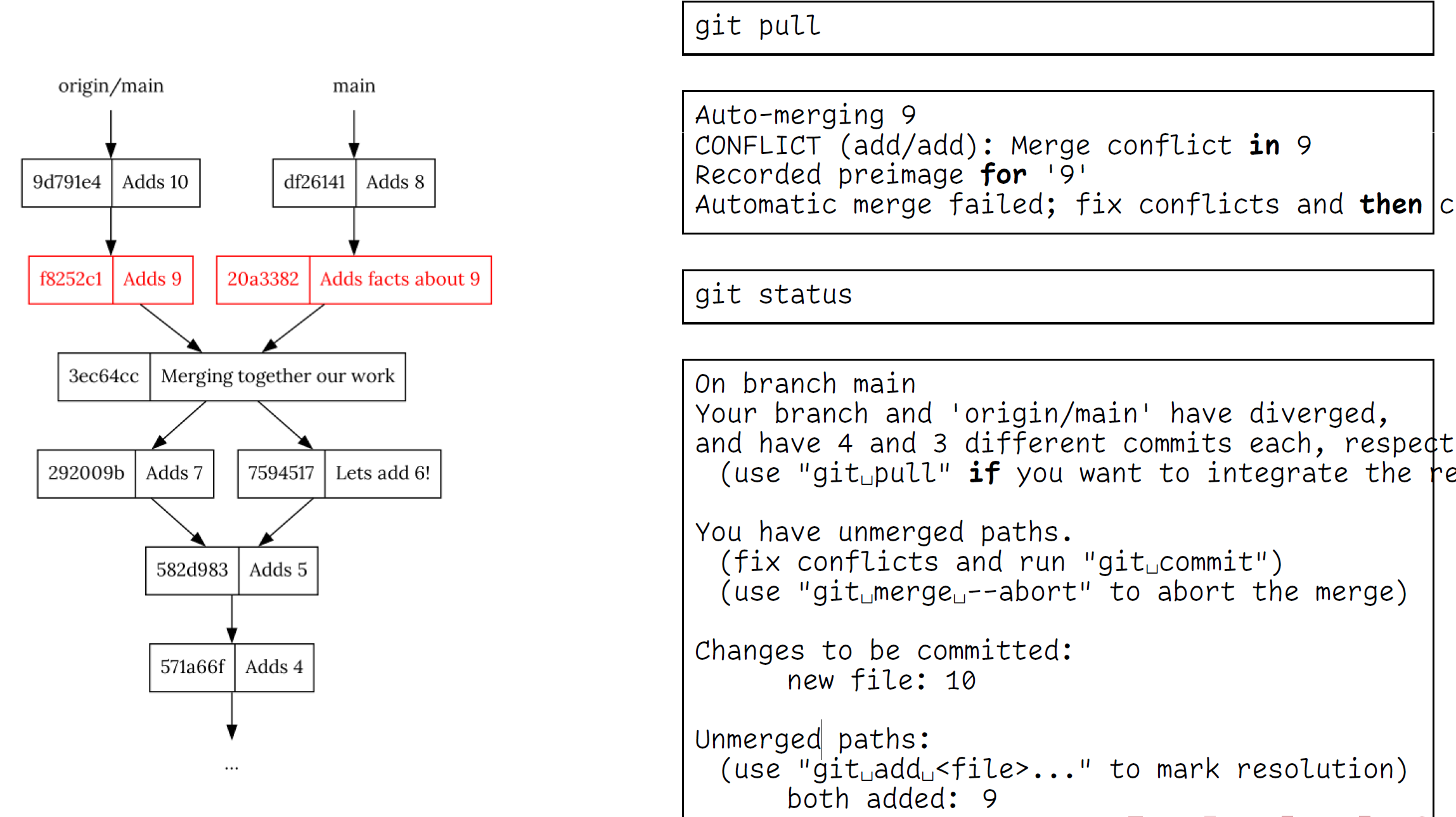
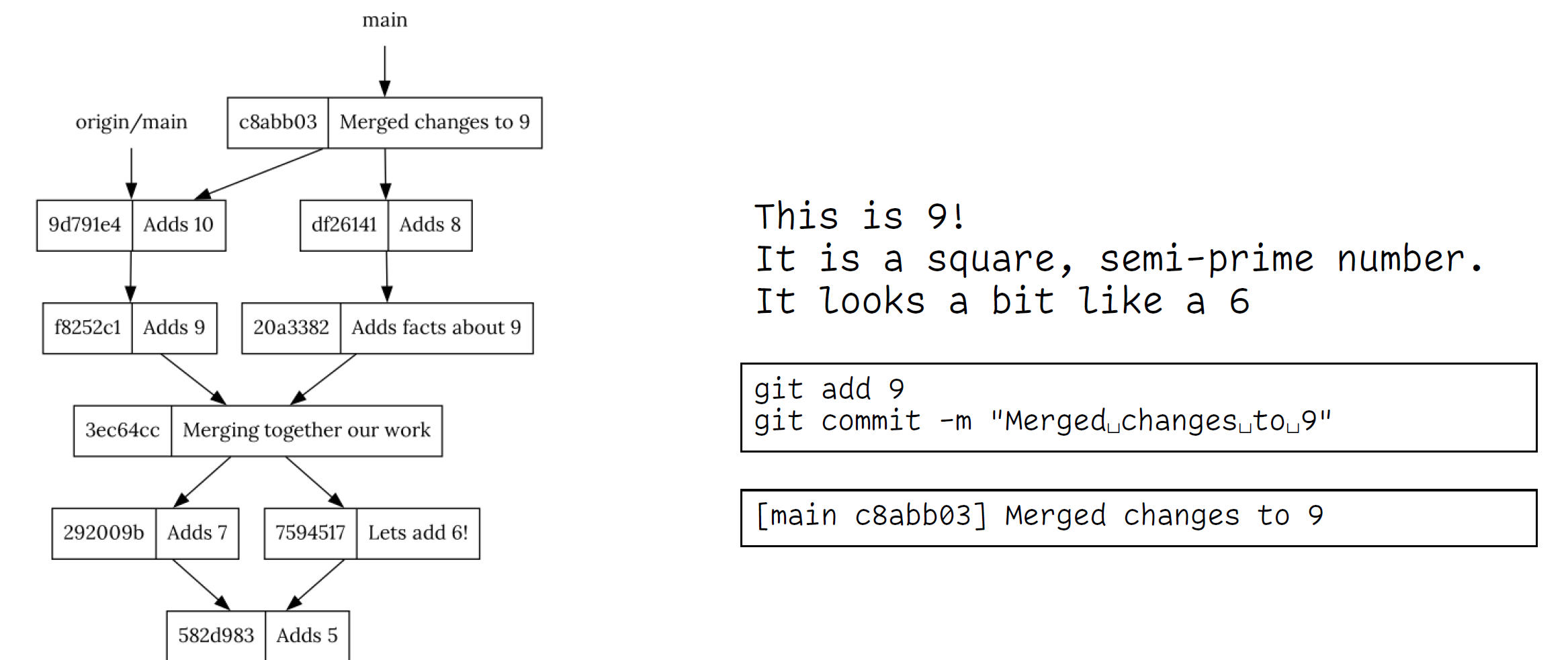
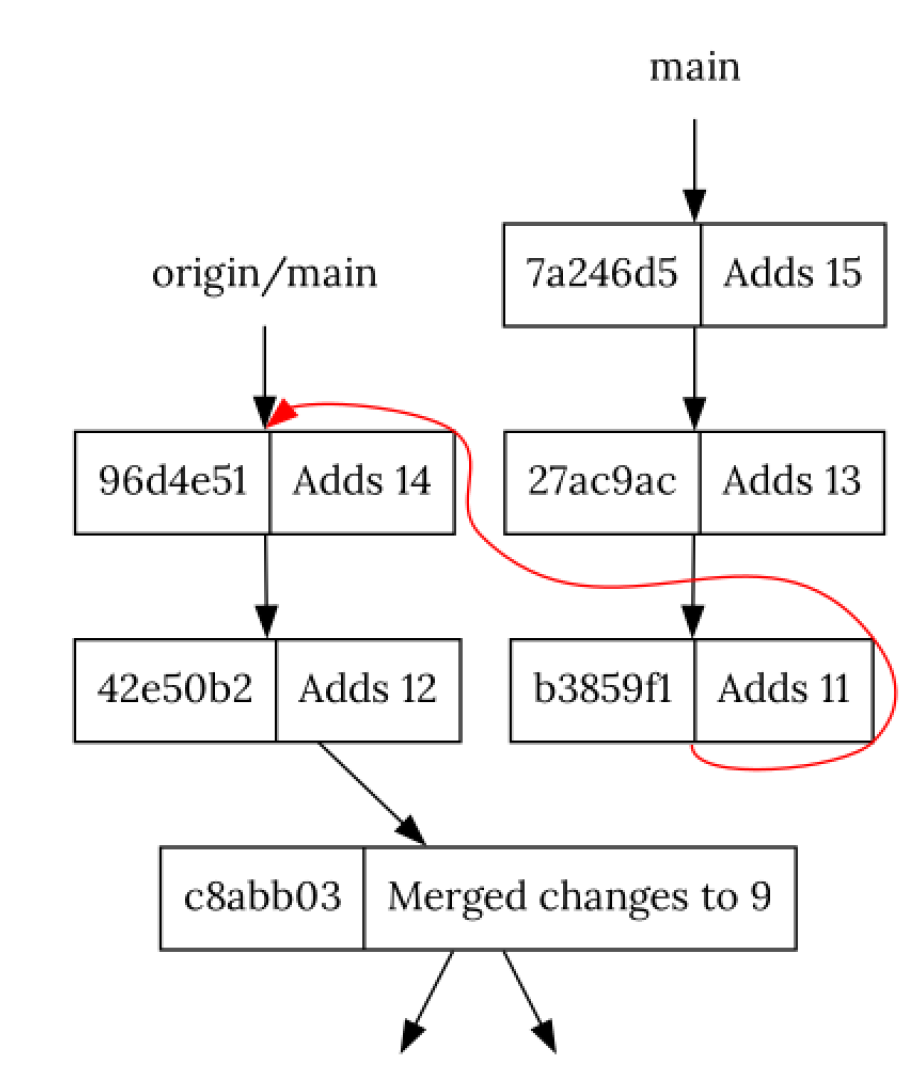
**git push --set-upstream origin develop**

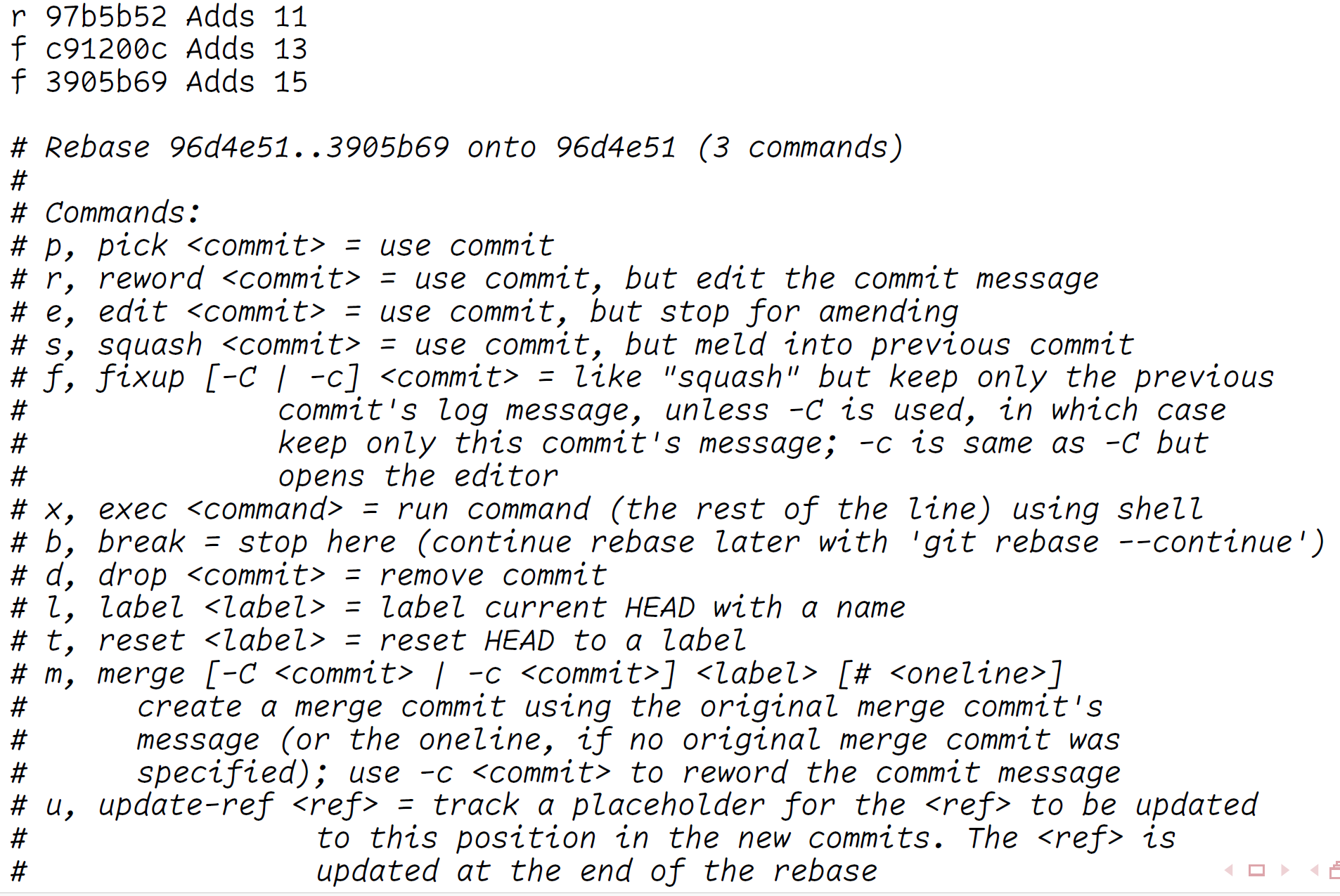
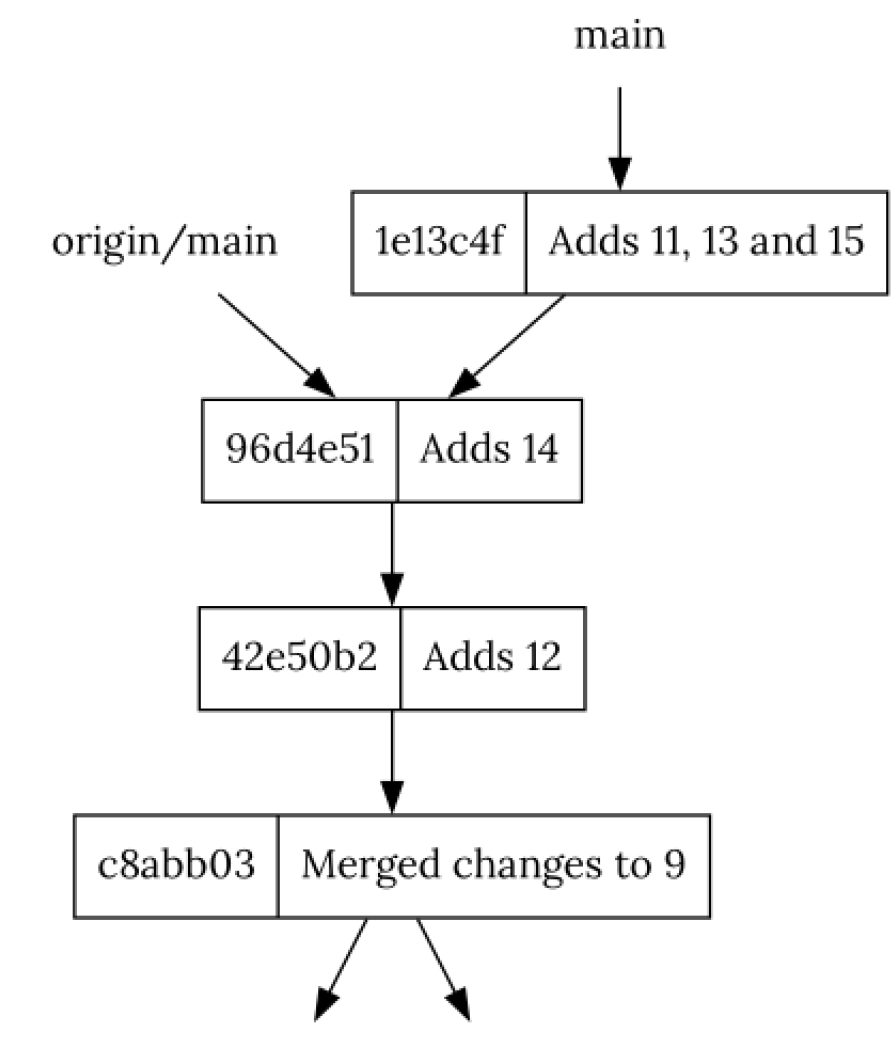
This adds an "upstream" entry on the local **develop** branch to say that it is linked to the copy of your repository called **origin**, which is the default name for the one you cloned the repository from.

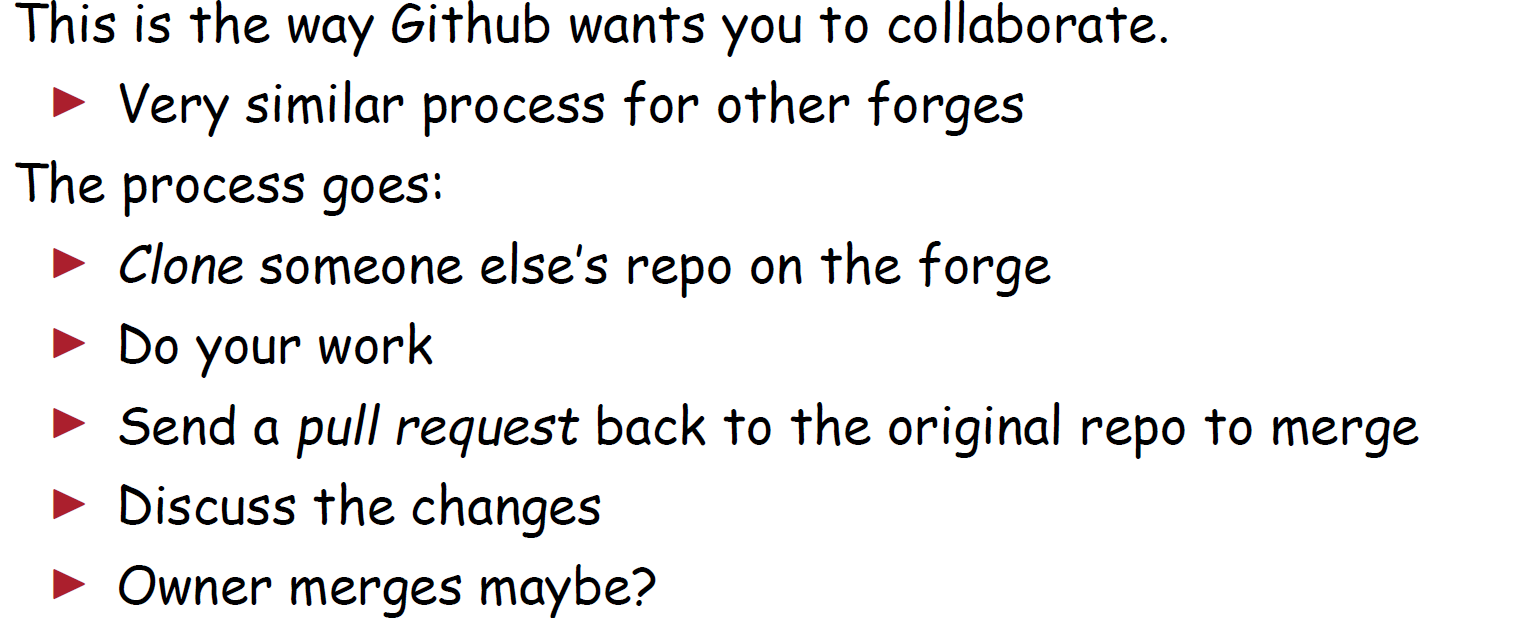
The origin doesn’t know about our merge! I We need to send our changes up to it：git push；This does not change our collaborators’ code tree!

   git rebase --onto origin/main

git **cherry-pick** <commitHash>





git config core.excludesFile：apply .gitignore tto *your* machine

工作流程：

建立开发分支git checkout -b develop；git push --set-upstream origin develop

建立每个人的特性分支git checkout -b NAME；git push --set-upstream origin NAME；git push

和开发分支合并git fetch；git checkout develop；git status；git merge NAME ；git push

不同人不同分支不同文件修改后push：git rebase develop；git push --force origin BRANCHNAME

提交并推送您的更改。

如果有必要，将您的特性分支重新建立在开发分支上。

创建拉取请求。

如果有必要，参与讨论或审查，并作出额外的提交，以解决其他开发人员提出的问题。

有人——通常不是创建拉取请求的开发人员——批准它，在开发（或主）中创建合并提交。

1. **Build Tools**

POSIX Make；GNU Make/gnumake/gmake

greeter: greeter.c library.o library1.o

$CC $CFLAGS greeter.c library.o library1.o

To build greeter you’ll need the files: greeter.c, library.o and library1.o

Rebuild any of them if their source is newer

To do the build you need to run $CC $CFLAGS greeter.c library.o library1.o

This line is indented with a **tab!**

CC=gcc；CFLAGS=-O2：Set variables to these values；**No spaces around the =**；$(CC)

以#开头的是注释，会被make命令忽略。

make执行时，默认执行第一条规则。*or* it’ll run .**default**：.default greeter

clean:

rm -f m.txt

rm -f x.txt

**clean**规则与我们前面编写的规则有所不同，它没有依赖文件，因此，要执行clean，必须用命令make clean

%.o: %.c

$CC $CFLAGS -c $^ -o **$@**

%.o: %.c To get /something/=.o= you need a /something/=.c=

$^ The entire dependency list；$@ The target output；$< 表示第一个依赖项

**all** that buildseverything；**install** that installs your into path

make内置了隐式规则（Implicit Rule），即遇到一个xyz.o时，如果没有找到对应的规则，就自动应用一个隐式规则：

xyz.o: xyz.c

cc -c -o xyz.o xyz.c

CC=clang make all；Now the user can override them with an environment variable

%.png: %.dot **#**Use dot command to convert a .dot file into a .png file. $< refers to %.dot

dot -Tpng $< -o **$@**

flowcharts=$(**patsubst** .dot,.png,$(wildcard figures/\*.dot)) **#**substitute all .dot into .png

paper.pdf: paper.tex $(wildcard figures/\*.png) ${flowcharts} **#** **wildcard**通配符

pdflatex paper #use LaTeX source file，.png，.png generated from .dot

Make doesn’t know anything about the code its compiling；just pattern matching on extensions and access times

Commonlisp：ASDF and Quicklisp；Go：Gobuild；Haskell：Cabal；Java：Ant, Maven, Gradle…；JavaScript：NPM；Perl：CPAN；Python：Distutils and requirements.txt；R：CRAN；Ruby：Gem；Rust：Cargo；LATEX：CTAN and TeXlive

Maven：Configured in XML；*Needlessly* verbose；Fairly standard and available everywhere

mvn archetype:generate \

-DgroupId=uk.ac.bristol.cs \

-DartifactId=hello \

-DarchetypeArtifactId=maven-archetype-quickstart \

-DinteractiveMode=**false**

maven-archetype-quickstart is a popular archetype for generating: ./src/main/java/org/example/App.java；./src/test/java/org/example/AppTest.java；pom.xml. org. example=groupId

XML：Format for writing trees that can be parsed by a computer *and* a human；Basically a generalized form of HTML；schema defines what all the tags mean

<dependency>

<groupId>org.antlr</groupId>

<artifactId>antlr4-runtime</artifactId>

<**version**>4.13.1</**version**>

</dependency>

mvn **package**；mvn **test**；mvn **install**（install the JAR）；mvn **clean**

**lab**

install 到 /usr/local/bin需要root權限；

You could install it somewhere else. rerun ./configure with --prefix=${HOME}/.local/

apt-file search <name of file> 可以找到所需的library

python：pip3 install --user mistletoe

您可以使用pip而不使用sudo，通过传递--user选项，它将包安装到主目录（~/.local）中的文件夹中，而不是通常需要root权限的/usr中。

Python有一种叫做venv的机制，它允许你创建一个由用户拥有的虚拟Python安装

PIP freeze | tee requirements.txt将列出您使用的所有软件包及其版本，并将它们保存在一个名为requirements.txt的文件中。

PIP install -r requirements.txt将重新安装它们！

Java：javac编译器将源文件（.java）转换为.class；jar工具将.class打包成.jar；java命令运行.class或.jar。Java运行时环境（JRE）只包含Java命令，许多操作系统允许您双击jar文件（至少包含称为清单的特殊文件），以便在JRE中运行它们。JDK （Java Development Kit）包含javac和jar工具以及JRE。maven是一个Java包管理器和构建工具。它不是Java发行版的一部分，因此需要单独安装。

mvn archetype:generate如果您得到“not found”错误，那么很可能maven bin文件夹不在您的路径上。[*export PATH="$PATH:..." ]and preferably put that line in your ~/.profile*

<scope>test</scope>声明该框架表示它仅用于测试

<plugins>部分包含maven用来编译和构建项目的插件。这个部分不是强制性的，但是包含它是为了将插件“锁定”到一个特定的版本，这样如果发布了一个新版本的插件，它不会改变构建的工作方式。<artifactId>exec-maven-plugin</artifactId> [mvn exec:java]

您进行编辑，然后运行mvn compile test exec:java重新编译，运行测试，然后运行程序。

[mvn test] runs the tests in src/test/java

[mvn package] creates a jar file of your project in the target/ folder.

在与pom.xml相同的文件夹中创建一个.gitignore文件，并将target/行添加到其中

1. **Debugging**

AddressSanitizer-Adds extra debugging checks

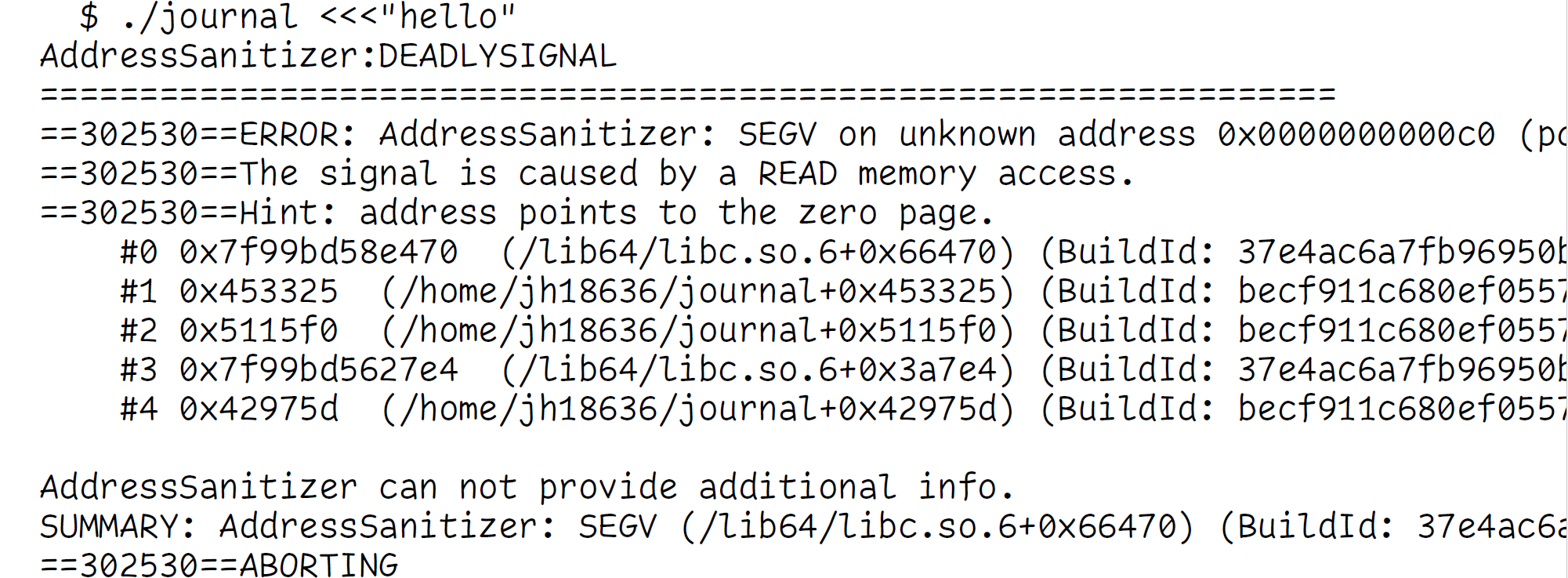
from the LLVM project (clang) to give more information about crashes **at runtime**

Your program will use more memory and be slower；easier to hack

clang journal.c -fsanitize=address -o journal

由于分段错误（SEGV）导致的；程序试图读取地址0xC0的内存，访问这样一个低地址通常发生在解引用NULL指针或无效指针时。

调用libc.so. 6，发生故障的地址如0x453325、0x5115f0等处具有函数



Other sanitizers exist:

UndefinedBehaviorSanitizer-spots bad compilier practice

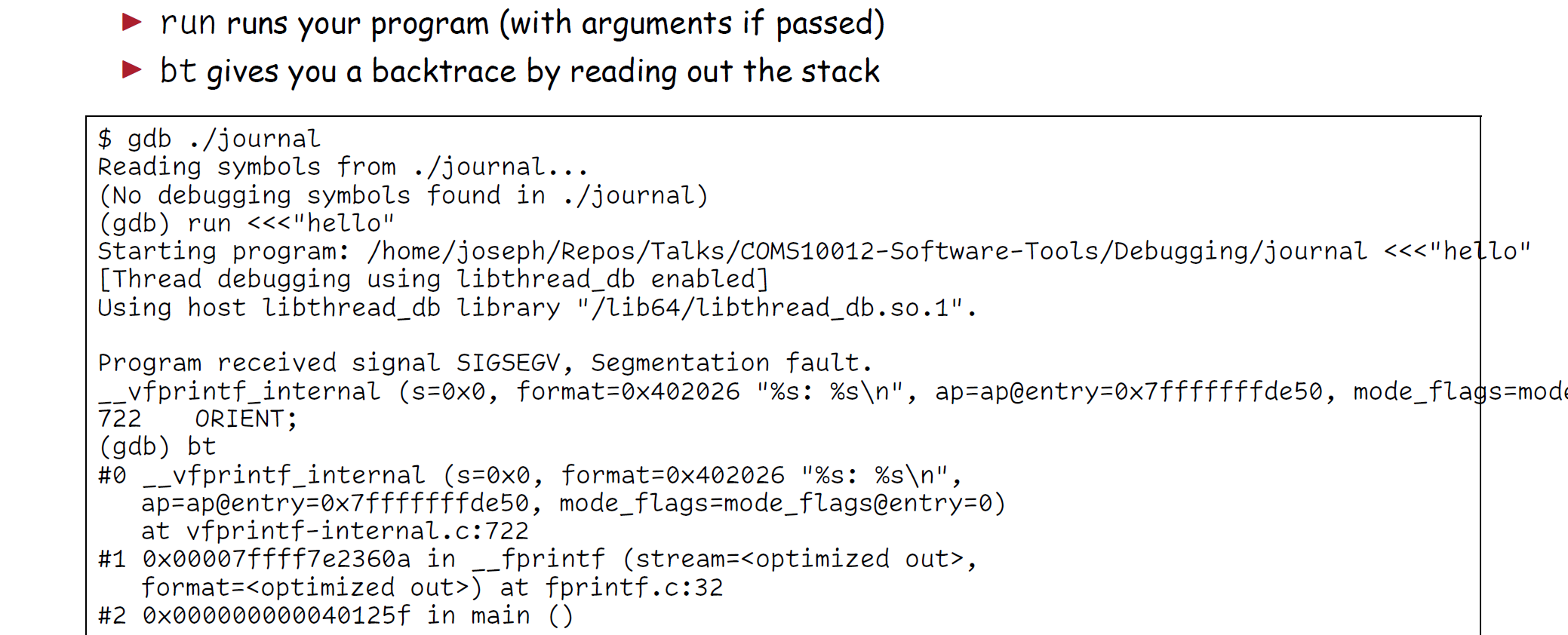
ThreadSanitizer-spots bad multithreading practice

LeakSanitizer-spots bad memory management

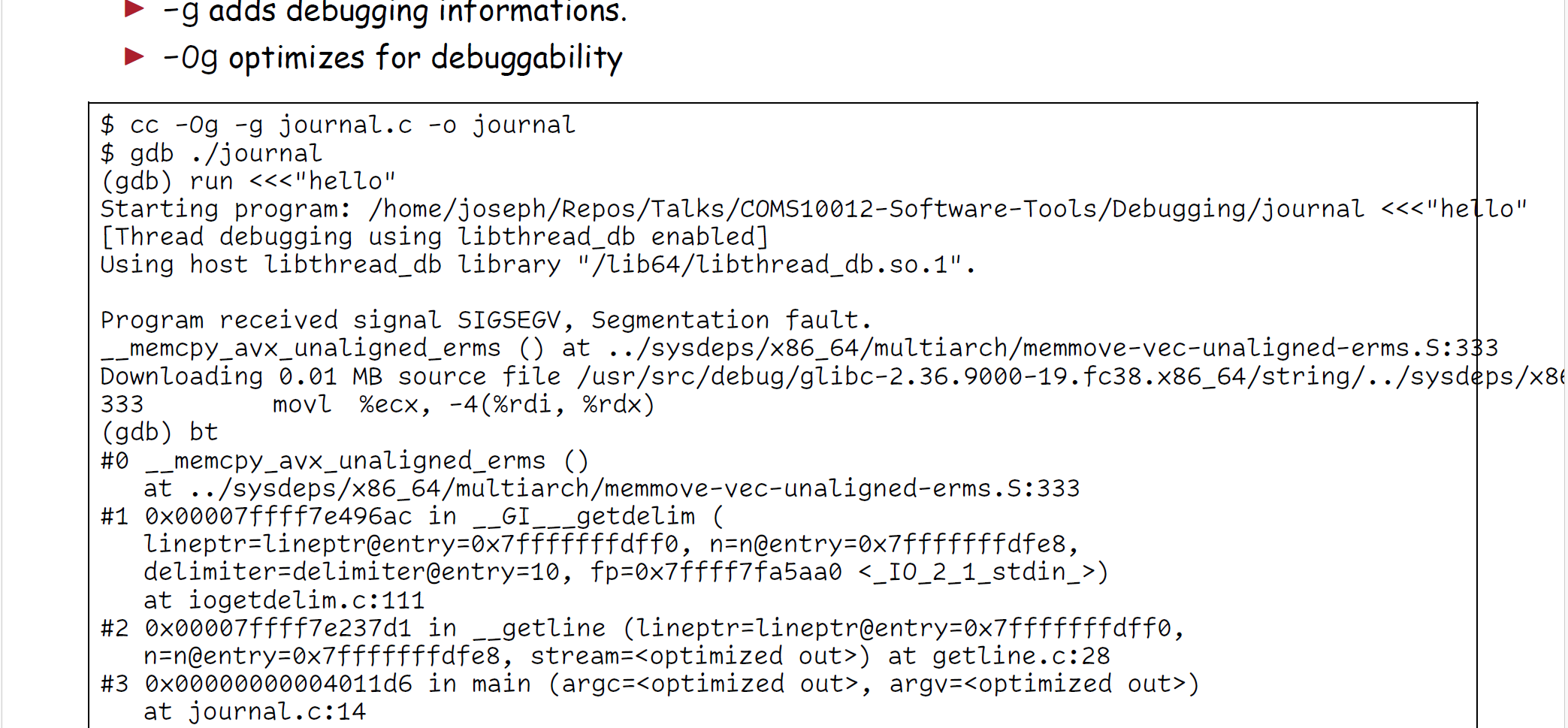
GDB-the GNU Debugger

Powerful: Can step through a program at an assembly level；Can watch registers and the stack change line by line；Can program scripts and to react when certain things happen；Can debug systems remotely；Ported to every sort of computer you could wish for

Unintuitive：It has a built in GUI but you’ll wish it didn’t；Lots of unintuitive single letter commands；Cryptic output；But its the standard debugger



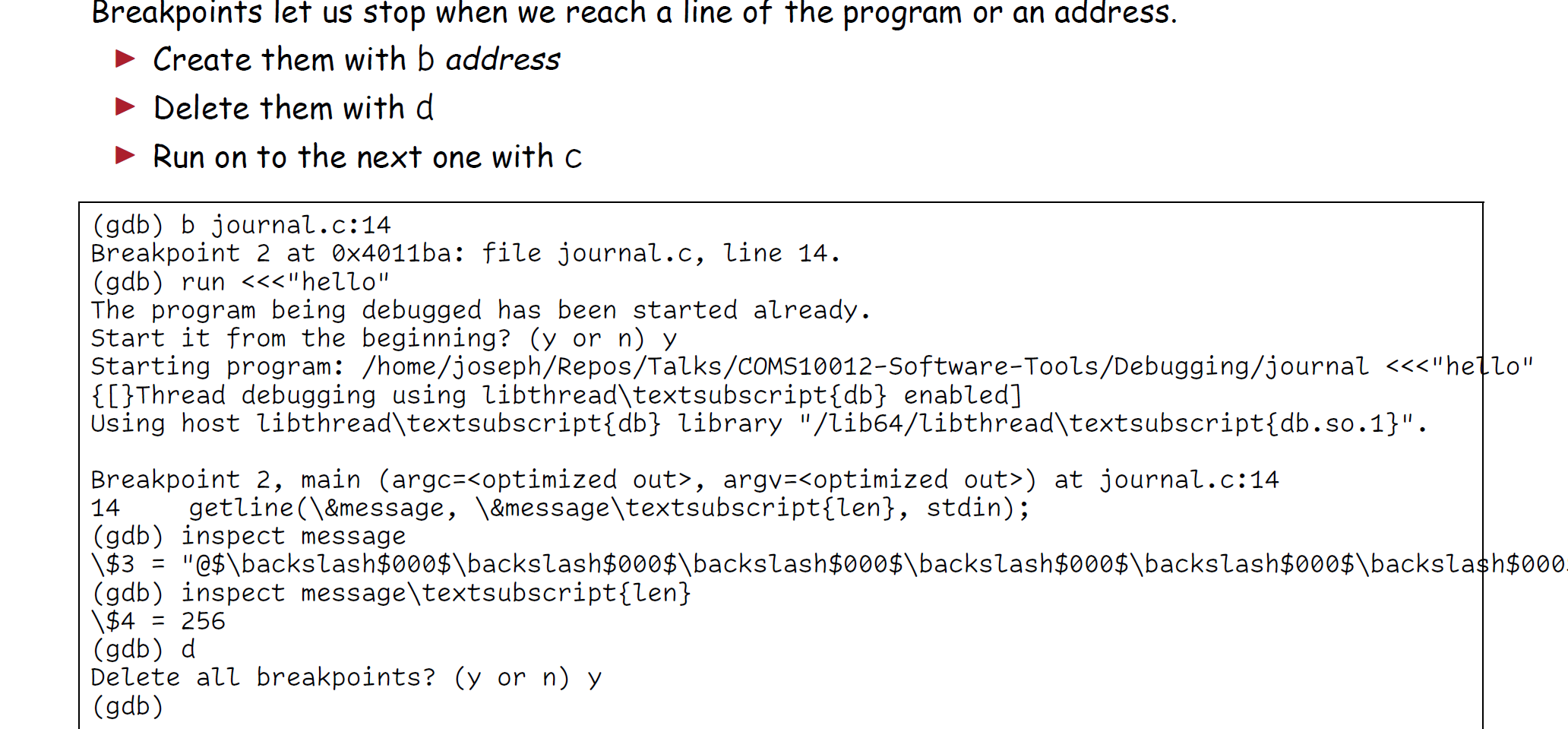
Fprintf，A FILE\* pointer was not initialized。the first parameter s=0x0 indicates that s (the file stream) is a NULL pointer.



提供给getline的参数中存在无效的指针或未初始化的内存

崩溃源于你在journal.c:14的main函数

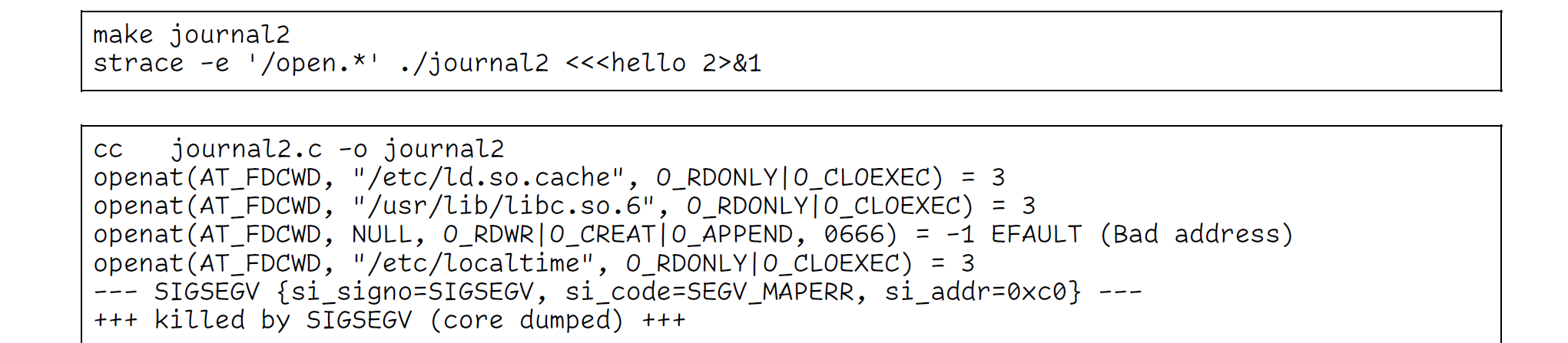
ssize\_t getline(char \*\*lineptr, size\_t \*n, FILE \*stream); 如果在调用之前将\*lineptr设置为NULL并且将\*n设置为0，则getline（）将分配一个缓冲区用于存储该行。或者，在调用getline（）之前，\*lineptr可以包含一个指向malloc(3)分配的缓冲区\*n字节的指针。



message is initialized with a valid memory location, but its contents seem uninitialized (represented as non-printable characters like @ followed by null bytes). 應Lineptr=null,len=0

gdb Info获取有关寄存器或变量或其他任何东西的信息；x检查变量/指针；disas查看您正在运行的汇编代码；help

Strace-Systemcall-tracer；OpenBSD ktrace and kdump；MacOS/FreeBSD dtruss and dtrace

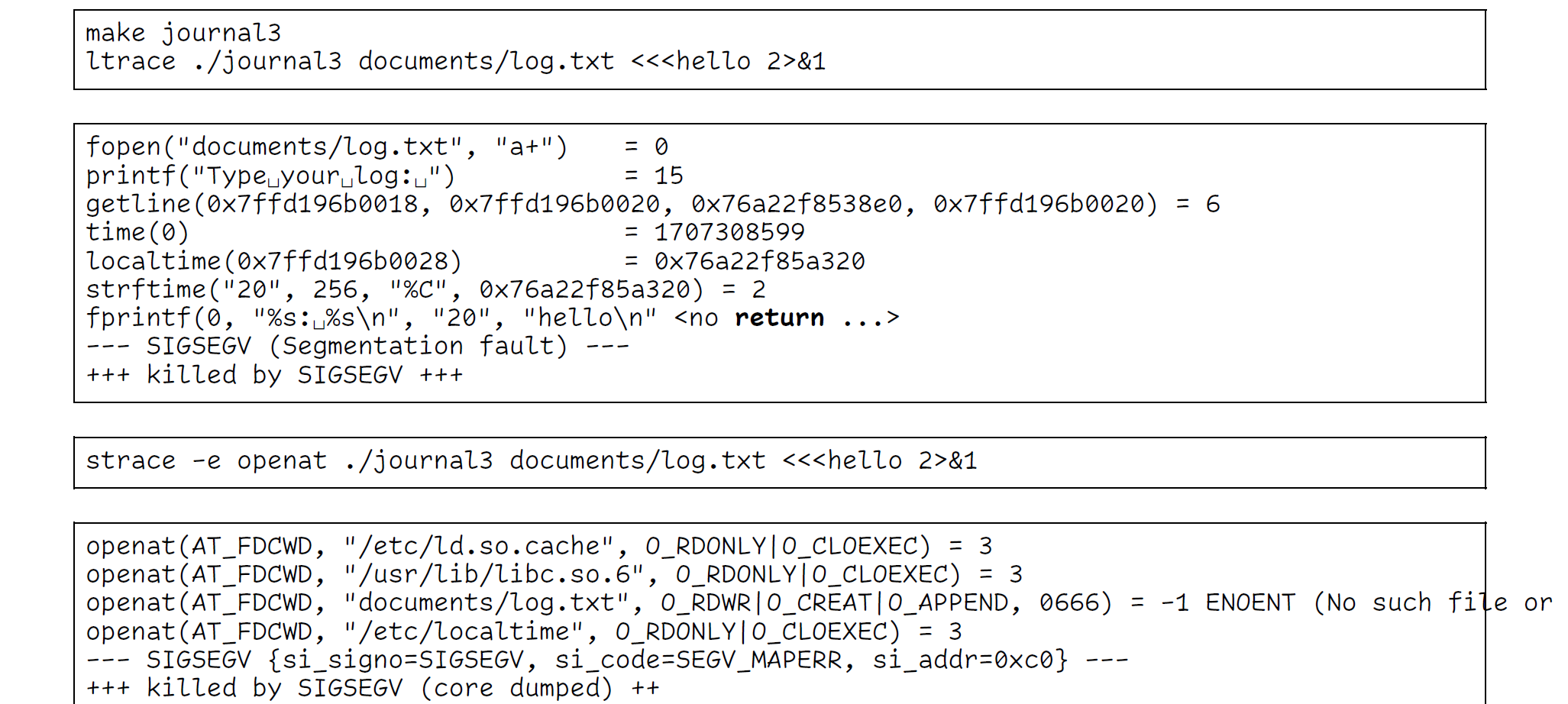


the dynamic linker is loading the shared libraries needed by the program, such as the C standard library (libc.so.6).

a call to openat with NULL as the file path, which is invalid.（未传入filename）

The program crashes due to a segmentation fault caused by accessing an invalid memory address (0xc0).

Ltrace-Library tracer

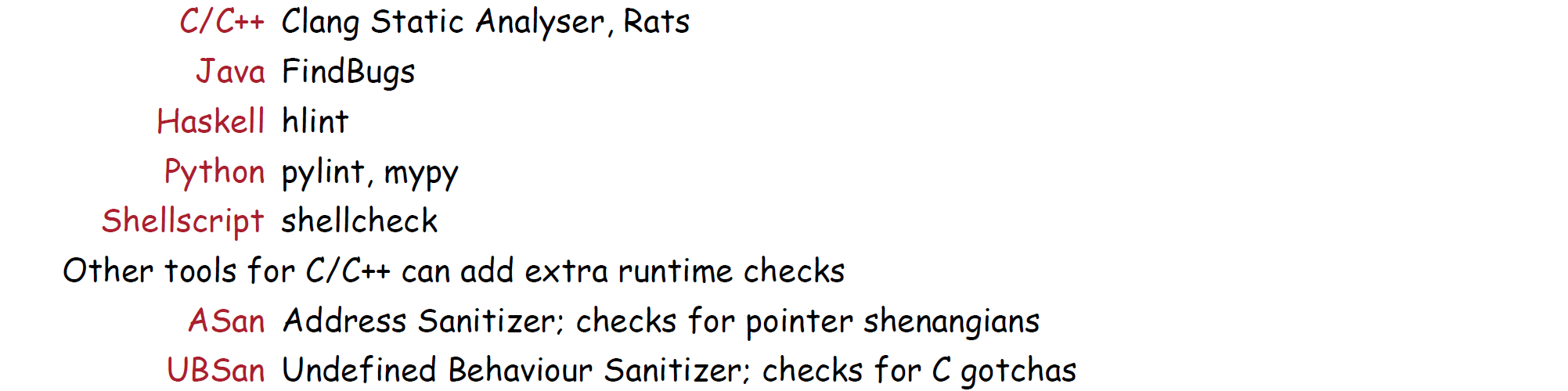
（文件不存在

Valgrind-Memory error detector

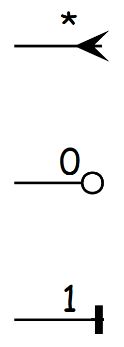


The **120 bytes lost** indicates a **definitely lost allocation**. This is a clear memory leak caused by failing to free some memory during the program's execution.

The **472 bytes still reachable** indicates memory that wasn't freed but is not considered a leak because it can still be accessed.



1. **Sql**

**Server** based :: MySQL, Oracle SQL, MariaDB, etc

These run on a server and provide distributed access to a single database

Good if you want to keep your database separate from your application

Running a webapp；Need multiple people able to connect at the same time

**File** Based :: SQLite, DuckDB, etc

Run in your application and provide structured storage

You’re building a mobile app；need to store data

行保存数据，表中每个项目一行，表内没有顺序（行应该是独立的）

列称为属性，每个属性描述了数据库中该行的一些信息

数据库包含许多表，属性可以引用其他表中的数据

A ***candidate*** *key* is a *minimal* set of attributes needed to uniquely refer to it.

The ***primary*** *key* for an entity is the key we use.

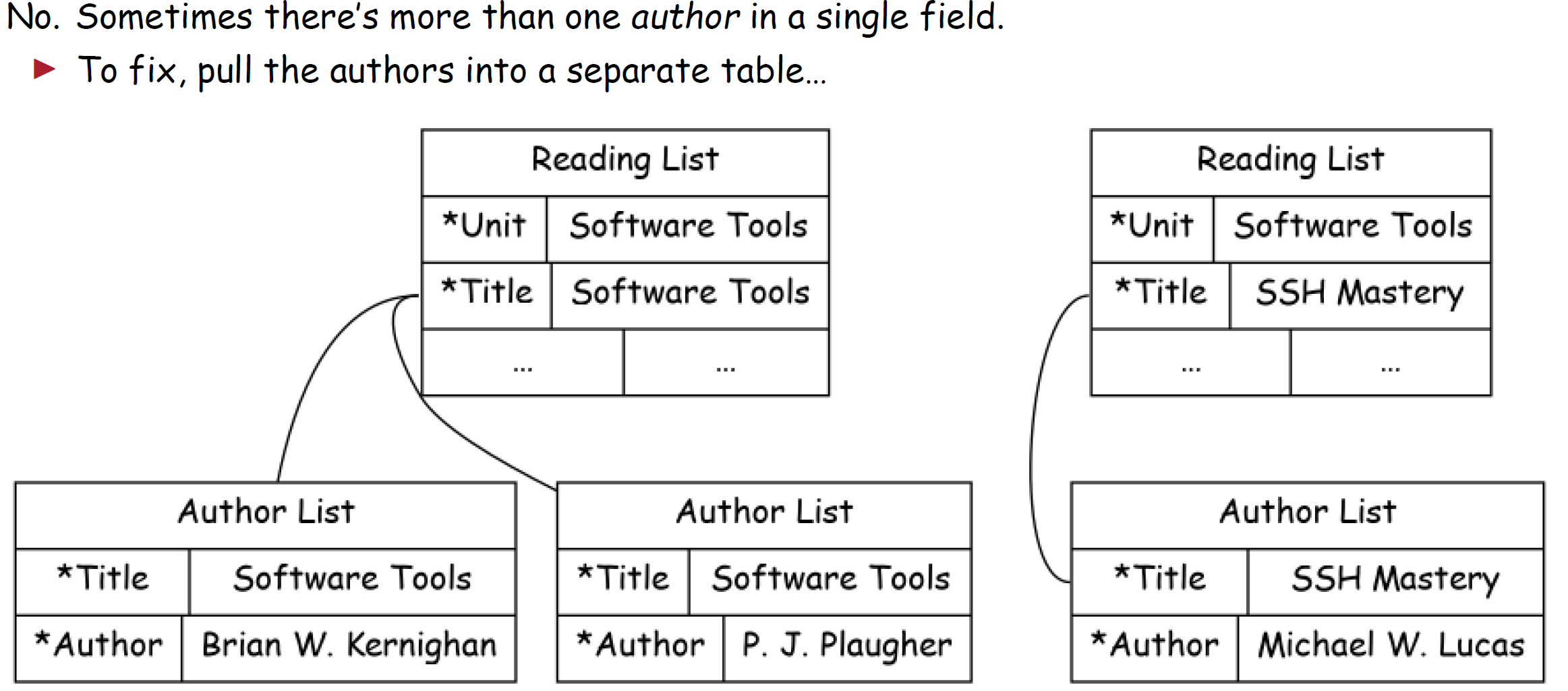
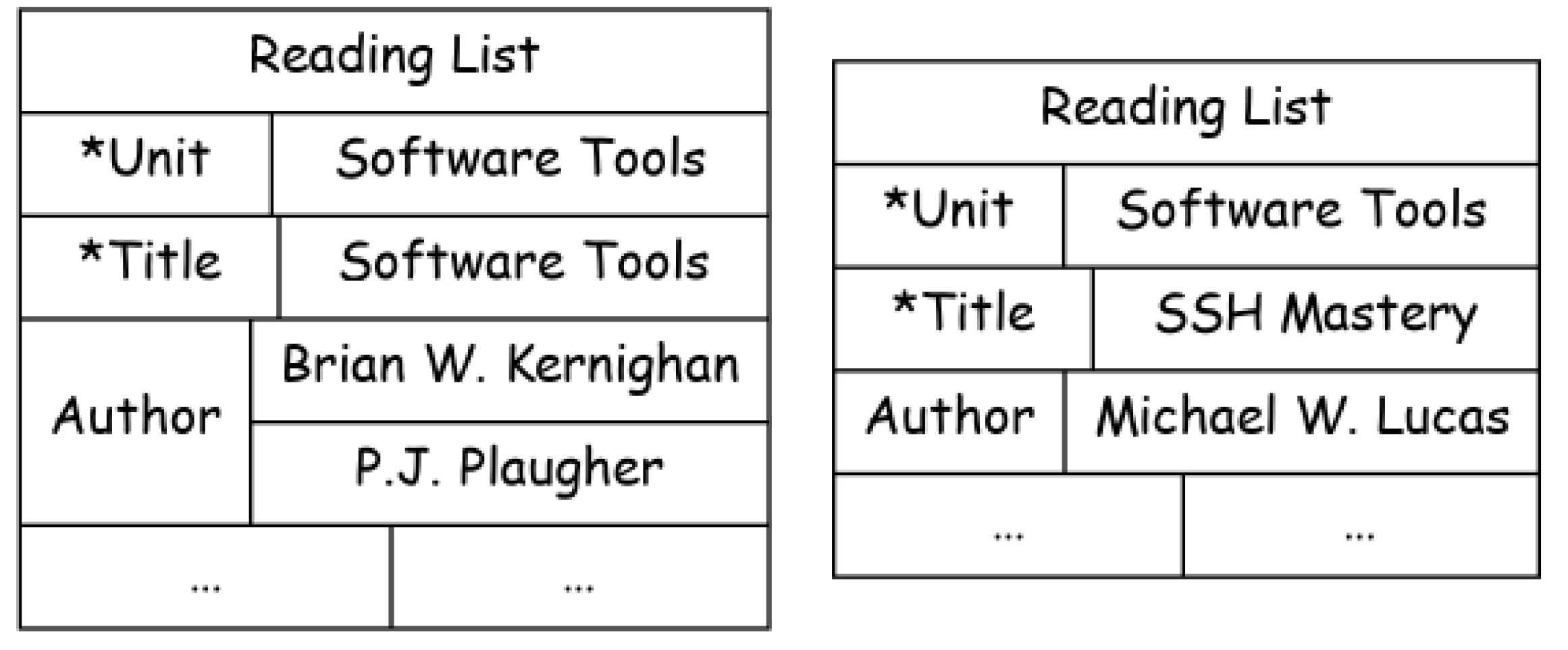
If a key contains multiple attributes it’s called a ***composite*** *key.*

If a key is a meaningless ID column you added just for the sake of having a key it’s called a ***surrogate*** *key*.

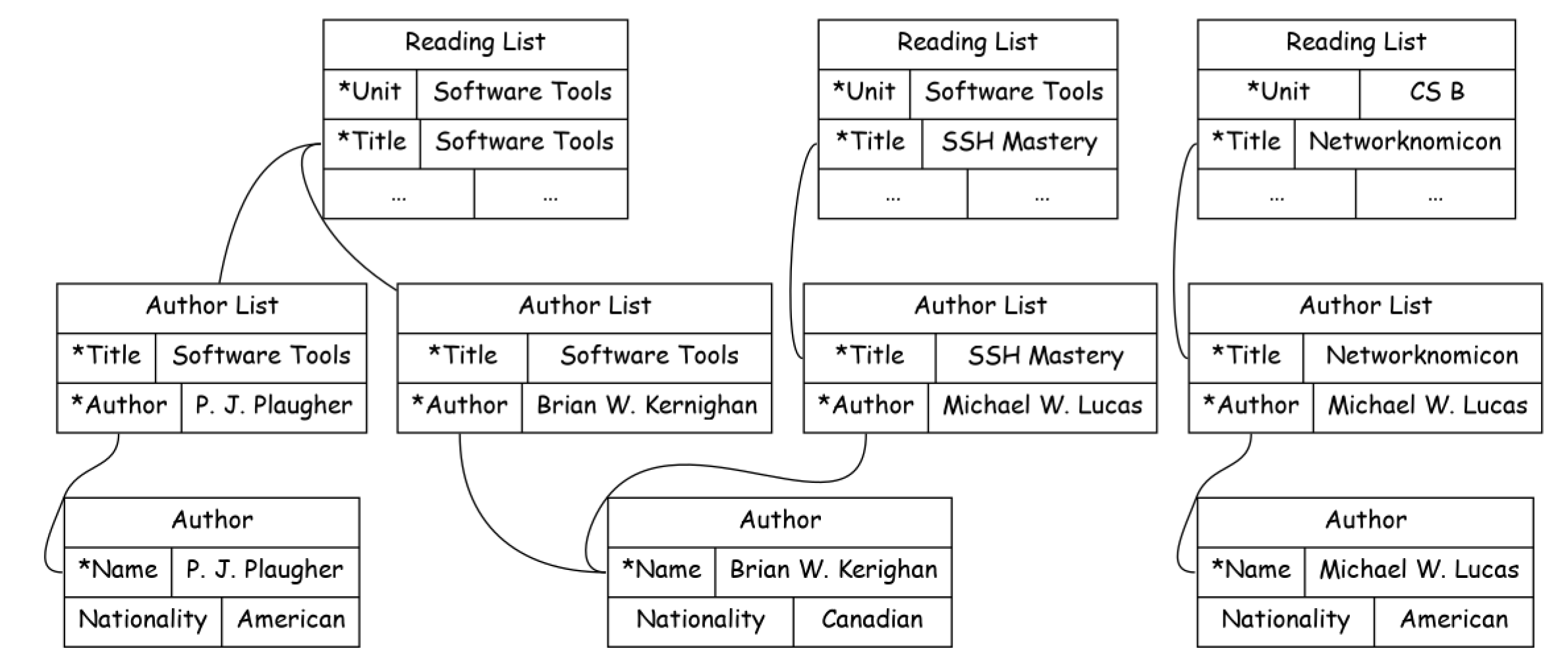
**Relational modelling** is a tool for thinking about how to decompose relationships between things into tables.

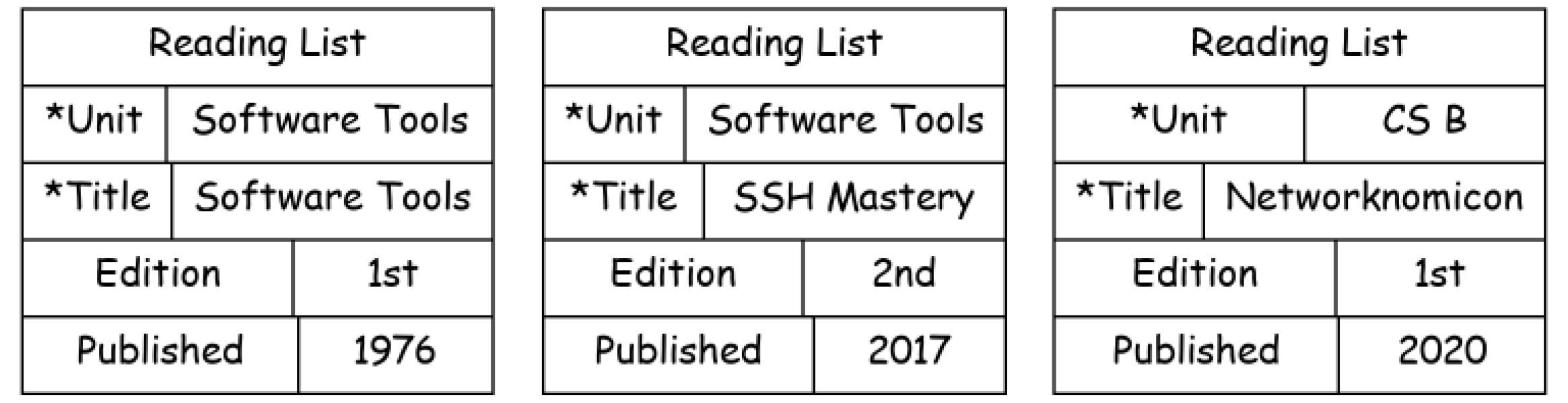
NF：Avoid having to add and update multiple database entries；Make it easy to delete stuff；Make it easy to find things

1NF：each field can only contain one value



2：All of the non-key attributes must depend on the whole key

*authors nationality* depends on the *author* but not the *title* of the book, so not the whole key

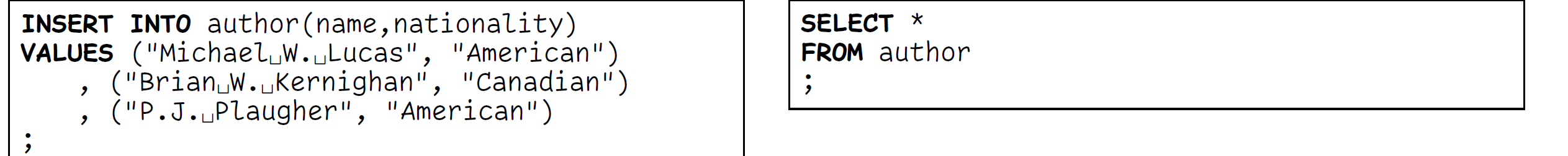
3：No non-prime attribute is transitively dependent on the primary key （It‘s in 2NF because *edition* and *published* both depend on the key. But the year published *could* depend on the edition and the *title*.）

If you ensure that there can never be a key that wouldn’t meet 3NF then its 3.5NF



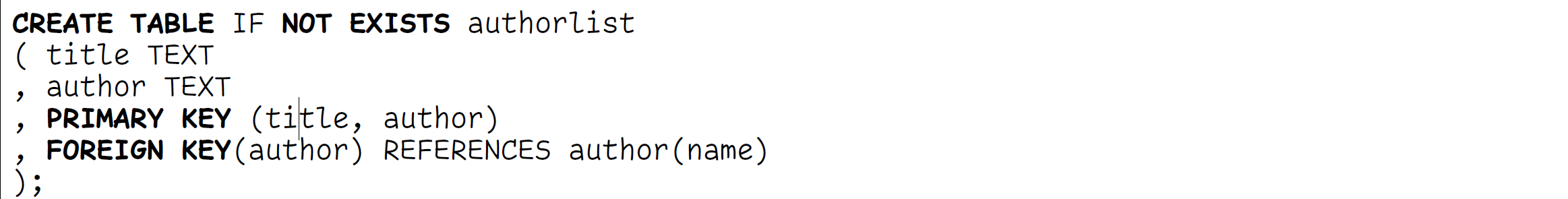
INT / INTEGER；TEXT；BLOB binary blobs of data (could be anything)；REAL floating point numbers；DECIMAL(5,2) a decimal number of 5 digits(2 of which are after the decimal point)；CHARACTER(10) / VARCHAR(10) A string of 10 characters or up to 10 characters.

DATE / DATETIME；BOOLEAN



NULL wreaks havoc when you come to JOIN tables to each other

NOT NULL；UNIQUE；CHECK () run a check that a value meets a condition；PRIMARY KEY implies NOT NULL and UNIQUE



**SELECT** Name **FROM** Artist **LIMIT** 5; **SELECT COUNT**(Name) **AS Number FROM** Artist;

**SELECT** Name **FROM** Artist **WHERE** Name IS "Chappell Roan"; **SELECT** Name **FROM** Artist **WHERE** Name **LIKE** "Avril %";

**SELECT** Name **FROM** Artist **WHERE** Name **LIKE** "% zep%l%in%";

**SELECT** Album.Title, Artist.Name **FROM** Album **JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **WHERE** Artist.Name **LIKE** "Led Zeppelin";

**SELECT COUNT**(Album.Title) **AS** Albums, Artist.Name **FROM** Album **JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **WHERE** Artist.Name **LIKE** "Led%" **GROUP BY** Artist.Name;

**SELECT COUNT**(Album.Title) **AS** Albums, Artist.Name **FROM** Album **JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **GROUP BY** Artist.Name **ORDER BY** Albums **DESC** **LIMIT** 10;

**SELECT COUNT**(Album.Title) **AS** Albums, Artist.Name **FROM** Album **JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **GROUP BY** Artist.Name **HAVING** Albums >= 5 **ORDER BY** Albums **DESC**;

COUNT()；SUM()；MAX()；MIN()；AVG()

JOIN=INNER JOIN，需要两张表中都有的内容，If we want to include them we need an OUTER JOIN=FULL OUTER JOIN

LEFT OUTER JOIN，there’s something in a but nothing in b to join it to… then leave a NULL；RIGHT OUTER JOIN

**SELECT** "Artists with no album" **AS** Description, **COUNT**(Artist.Name) **AS Count** **FROM** Album **RIGHT OUTER JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **WHERE** Album.Title IS **NULL**

**UNION**

**SELECT** "Albums with no artist" **AS** Description, **COUNT**(Album.Title) **AS Count** **FROM** Album **LEFT OUTER JOIN** Artist

**ON** Album.ArtistId = Artist.ArtistId **WHERE** Artist.Name IS **NULL**

**CREATE TEMPORARY TABLE** AlbumsPerArtist **AS SELECT COUNT**(Album.Title) **AS** Albums, Artist.Name **FROM** Album

**RIGHT OUTER JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **GROUP BY** Artist.Name;

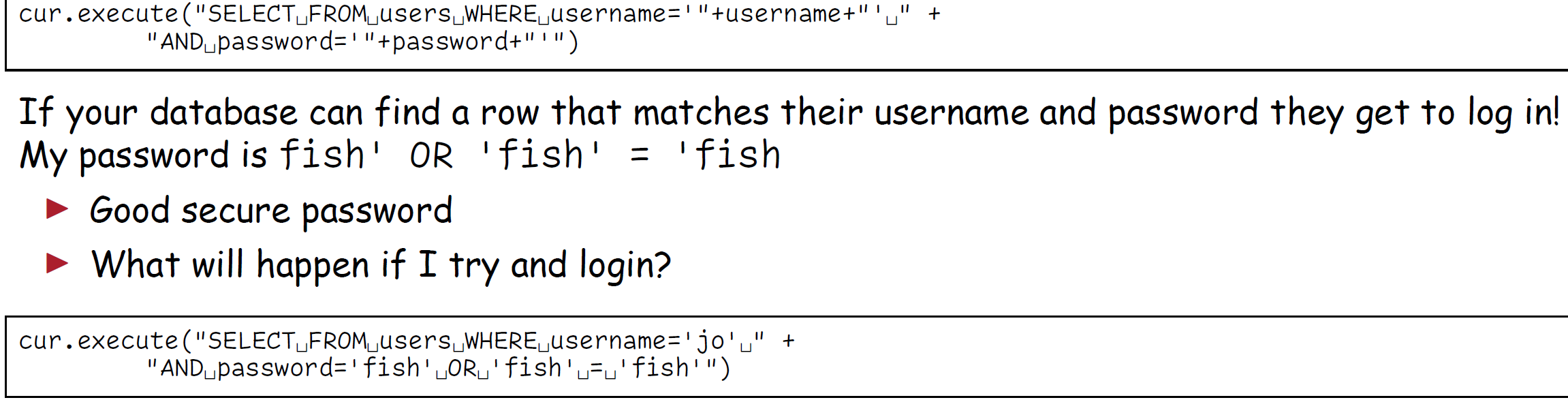
**SELECT SUM**(Albums) **as** Albums, **COUNT**(Name) **AS** Artists, **AVG**(Albums) **as** "Albums per Artist" **FROM** AlbumsPerArtist;

**SELECT SUM**(Albums) **as** Albums, **COUNT**(Name) **AS** Artists, **AVG**(Albums) **as** "Albums per Artist"

**FROM** (**SELECT COUNT**(Album.Title) **AS** Albums, Artist.Name **FROM** Album **RIGHT OUTER JOIN** Artist **ON** Album.ArtistId = Artist.ArtistId **GROUP BY** Artist.Name);

Python provides SQLite bindings through the sqlite3 library.

cur.execute("INSERT INTO Artist(ArtistId, Name) VALUES (?,?)", artist)



SQL Injection vulnerabilities

Transactions：Let me define a bunch of database queries as a single atomic operation；If *any* fail I can roll back all the changes and start again；Until I commit() my work nothing is saved

Java Rather than implementing SQL API’s for every database engine…Implement a general framework for database access，*JDBC；Database engines provide they’re own bridge to the JDBC API*

Go and fetch the driver you need (or use Maven) and stick it in your classpath

Datalog is a database language based on first order logic；

Simplified version of the logic programming language *Prolog；*

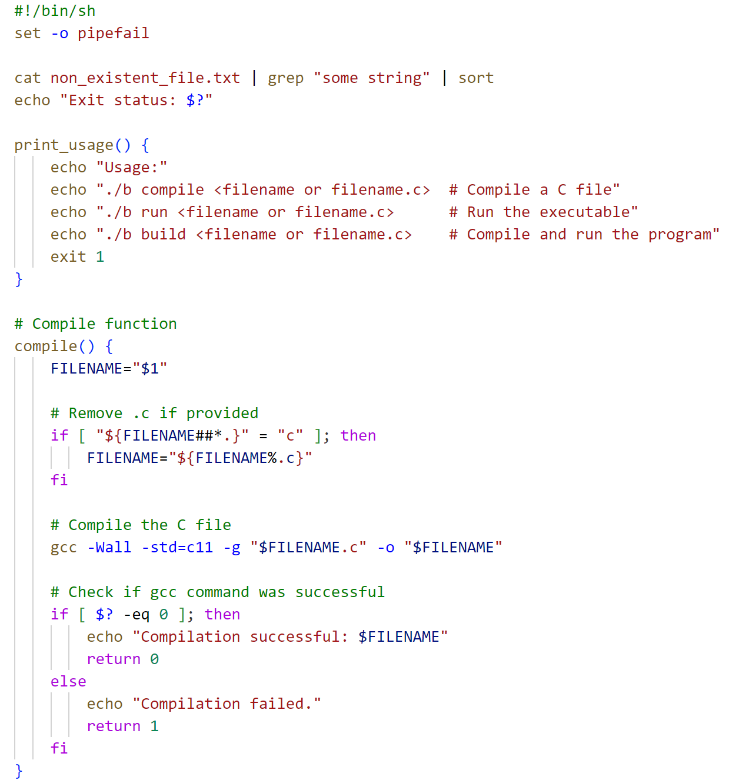
All facts/assertions are written as Horn clauses：predicate(arguments)；Variables are capitalized, constants are not；Rules separated with a :-，conjunctions with 逗号：canAuthorizeLeave(P,B) :- manages(P,B) manages(P,BB) :- manages(P,B), manages(B,BB)；All variables in the left hand of a rule *must* be referred to in the righthand side；Cannot express infinite sized sets (e.g. integers)：canRead(U,F) :- user(U), file(F), otherReadBitSet(F)；

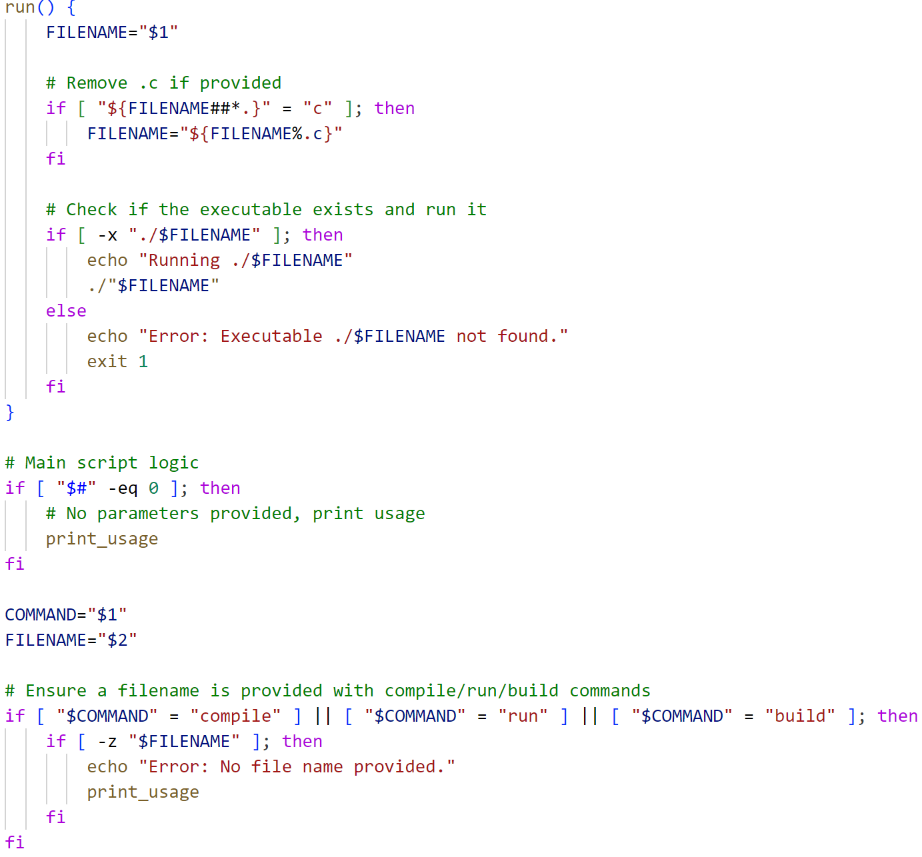
*Closed world assumption：*The database contains all the facts and rules you might ever need to prove something；If you *can’t* prove it with all the facts then it’s false；

Datalog Can represent trees；Can *provably* find all possibilities in a reasonable time；Can produce proofs *given* the right implementation；*With the right implementation* can find all values of a variable satisfying a query；*With extensions* can run quite complex queries

No infinite sets (no numbers)；*Research-grade* implementations

Program analysis；Access control；Machine learning and AI；Google/Facebook-scale data





All words containing the letter Q, capitalized：grep 'Q' /usr/share/dict/words

All words starting with the letter R, in either upper or lower case：grep -i '^r' /usr/share/dict/words

All words ending in j：grep 'j$' /usr/share/dict/words

The number of words containing the letter Q, ignoring case：grep -i 'q' /usr/share/dict/words | wc -l

The first five words containing the letter sequence 'cl'：grep 'cl' /usr/share/dict/words | head -n 5

All words containing the sequence "kp", but not "ckp"：grep 'kp' /usr/share/dict/words | grep -v 'ckp'

The last 15 words of exactly two letters：grep '^..$' /usr/share/dict/words | tail -n 15

All three-letter words with no vowels (aeiou)：grep '^[^AEIOUaeiou][^AEIOUaeiou][^AEIOUaeiou]$' /usr/share/dict/words

All words of exactly 7 letters, where the third one is an e and the word ends with "-ded"：grep '^..e.ded$' /usr/share/dict/words

Find all words that start with a P (whether capitalized or not), and contain at least four instances of the letter a：grep -i '^p[^a]*\*a[^a]\**a[^a]*\*a[^a]\**a$' /usr/share/dict/words

Find all dictionary words ending in 'ay' and replace 'day' with 'week'：

grep 'ay$' /usr/share/dict/words | sed 's/day/week/'

In the same selection as above, replace all words that begin with 's' with the word 'sway'.

grep 'ay$' /usr/share/dict/words | sed 's/^s.\*/sway/'

Duplicate the match after a space for lines containing 'day' (e.g., 'saturday' becomes 'saturday day')：

grep 'ay$' /usr/share/dict/words | sed 's/day/& day'

Turn lines ending in 'day' into "Xday or Xweek", where X is the part of the word before 'day'：

grep 'day$' /usr/share/dict/words | sed 's/\(.\*\)day/\1day or \1week/'

Flip words ending in 'way' or 'day', so 'someday' becomes 'day (some)' and 'speedway' becomes 'way (speed)'：

grep -E'(way|day)$' /usr/share/dict/words | sed 's/\(.\*\)\(way|day\)/\2 (\1)/'

Difference between s/a/e/ and s/a/e/g

echo "banana" | sed 's/a/e/'：benana；echo "banana" | sed 's/a/e/g'：benene

grep -o counts each occurrence of pattern, even if they appear multiple times on the same line.

grep -c counts lines containing at least one match, regardless of how many times pattern appears in that line.

encyclo(p|ae)dia

colou?r

^[a-zA-Z]{2}\d\s?\d[a-zA-Z]{2}$

parent(pugsley, gomez).

parent(wednesday, gomez).

parent(pugsley, morticia).

parent(wednesday, morticia).

parent(gomez, grandpa).

parent(fester, grandpa).

parent(morticia, grandma).

parent(sharron, debbie).

parent(pubert, gomez).

parent(pubert, morticia).

grandparent(X, Z) :- parent(X, Y), parent(Y, Z).

?- parent(X, sharron).

sibling(X, Y) :- parent(Z, X), parent(Z, Y), X != Y.

?- sibling(gomez, fester).

?- sibling(wednesday, wednesday).

cousin(X, Y) :- grandparent(Z, X), grandparent(Z, Y), not(sibling(X, Y)), X != Y.

?- cousin(gomez, morticia).

aunt\_or\_uncle(X, Y) :- parent(Z, Y), sibling(X, Z).

?- aunt\_or\_uncle(fester, X).

owner(F, P) :- principal(P), file(F).

canRead(P, F) :- owner(F, P).

canRead(P, F) :- othersCanRead(F).

saysCanRead(P1, P2, F) :- owner(F, P1), principal(P2), file(F).

canRead(P, F) :- saysCanRead(P1, P, F), owner(F, P1).

delegatesTo(P1, P2) :- principal(P1), principal(P2).

saysCanRead(P1, P2, F) :- delegatesTo(P1, P2), saysCanRead(P3, P1, F).

holds(P, R) :- principal(P), role(R).

saysCanRead(P1, R, F) :- owner(F, P1), holds(P2, R), principal(P2), file(F).

security\_level(unclassified).

security\_level(secret).

security\_level(topsecret).

level\_order(unclassified, secret).

level\_order(secret, topsecret).

clearance(P, L) :- principal(P), security\_level(L).

canRead(P, F) :- clearance(P, L1), security\_level(L2), L1 >= L2.

canWrite(P, F) :- clearance(P, L1), security\_level(L2), L1 <= L2.

canWrite(P, F) :- clearance(P, L1), security\_level(L2), L1 > L2.

canRead(P, F) :- clearance(P, L1), security\_level(L2), L1 < L2.

1. List the names of all parties that stood in the election, ordered alphabetically by name.

#+begin\_src sqlite <<preamble>> SELECT DISTINCT Elections.Party.name FROM Elections.Candidate JOIN Elections.Party ON Elections.Candidate.party = Elections.Party.id ORDER BY Elections.Party.name ASC ; #+end\_src

2. List the names of all parties that stood in the Bedminster ward.

#+begin\_src sqlite <<preamble>> SELECT DISTINCT Elections.Party.name FROM Elections.Candidate JOIN Elections.Party ON Elections.Candidate.party = Elections.Party.id JOIN Elections.Ward ON Elections.Candidate.ward = Elections.Ward.id WHERE Elections.Ward.name = “Bedminster” ORDER BY Elections.Party.name ASC ; #+end\_src

3. How many votes did Labour get in the Stockwood ward?

#+begin\_src sqlite <<preamble>> SELECT Elections.Candidate.votes FROM Elections.Candidate JOIN Elections.Party ON Elections.Candidate.party = Elections.Party.id JOIN Elections.Ward ON Elections.Candidate.ward = Elections.Ward.id WHERE Elections.Ward.name = “Stockwood” AND Elections.Party.name = “Labour” ORDER BY Elections.Party.name ASC ; #+end\_src

4. List the names, parties and number of votes obtained for all candidates in the Southville ward. Order the candidates by number of votes obtained descending (winner comes first).

#+begin\_src sqlite <<preamble>> SELECT Elections.Candidate.name , Elections.Party.name , Elections.Candidate.votes FROM Elections.Candidate JOIN Elections.Party ON Elections.Candidate.party = Elections.Party.id JOIN Elections.Ward ON Elections.Candidate.ward = Elections.Ward.id WHERE Elections.Ward.name = “Southville” ORDER BY Elections.Candidate.votes DESC ; #+end\_src

5. List the name, party and number of votes obtained for the winner only in the Knowle ward.

#+begin\_src sqlite <<preamble>> SELECT Elections.Candidate.name , Elections.Party.name , Elections.Candidate.votes FROM Elections.Candidate JOIN Elections.Party ON Elections.Candidate.party = Elections.Party.id JOIN Elections.Ward ON Elections.Candidate.ward = Elections.Ward.id WHERE Elections.Ward.name = “Knowle” ORDER BY Elections.Candidate.votes DESC LIMIT 1 ; #+end\_src

1. The university of Bristol is situated in the Cabot ward (ward names are not always distinct, but this one is). Find the names and codes of the CLU, region and country containing the Cabot ward (CLU = county level unit = “row in County table”).

#+begin\_src sqlite <<preamble>> SELECT \* FROM Census.County JOIN Census.Ward ON Census.Ward.parent = Census.County.code WHERE Census.Ward.name LIKE “Cabot” ; #+end\_src

2. Find the number of women in occupation class 1 (managers etc.) in the Cabot ward.

#+begin\_src sqlite <<preamble>> SELECT Census.Statistic.data FROM Census.Statistic JOIN Census.Occupation ON Census.Occupation.id = Census.Statistic.occId JOIN Census.Ward ON Census.Ward.code = Census.Statistic.wardId WHERE Census.Ward.name = “Cabot” AND Census.Occupation.name LIKE “%Manager%” AND Census.Statistic.gender = 1 LIMIT 5 ; #+end\_src

3. For the Stoke Bishop ward (E05002003), list the 9 occupation class names and the number of men in each occupation. Your table should have two columns called name and number. You can use the provided ward code, you do not need to join on the ward name.

#+begin\_src sqlite <<preamble>> SELECT Census.Occupation.name AS name , Census.Statistic.data AS number FROM Census.Statistic JOIN Census.Occupation ON Census.Occupation.id = Census.Statistic.occId JOIN Census.Ward ON Census.Ward.code = Census.Statistic.wardId WHERE Census.Ward.name = “Stoke Bishop” AND Census.Statistic.gender = 0 ; #+end\_src