# PART1-WEEK4-DEBUGGING

## Slides

**gdb:**

when compile:

- `-g` adds debugging information
- `-Og` optimizes for debuggability

`<<<input` runs your program with input

`b` set breakpoints

`c` continue after hitting a breakpoint

`bt` get a backtrace

`info` get information about registers or variables or anything else

`x` examine a variable/pointer

`help` get help

sequence：-g -Og编译；先`run <<<input`；之后 `bt`来回溯哪里有问题；然后`b filename.c: (bt断点的行号)`设置断点；再次run，然后看懂断点那里后，`inspect message`显示信息; `d` 删除所有断点

**strace:** The strace tool lets **you trace what systemcalls a program uses**

- On OpenBSD see `ktrace` and `kdump`
- On MacOS/FreeBSD see `dtruss` and `dtrace`
- strace lets you use `regexp` to filter what syscalls you look at
- 过滤：`strace -e '/open.*' ./journal2 <<<hello 2>&1`
- 找不同 `diff -u 相同文件名部分{2,3}.c`

**ltrace:** It traces **library calls**

`strace, ltrace, valgrind` and `gdb` will help deal with most of the bugs you encounter But so will good defensive programming strategies: Always check the return code of functions; Always check assumptions; Always fix your compiler warnings.

# Error point:

**sudo apt update/upgrade** → 对apt自身的操作，如果as root, 则不需要sudo

> **Q4**. Alice wants to delete the last commit they made on the `main` branch. What command or commands do they need to use?
>
> A. `git checkout main~1`
>
> B. `git clean`
>
> **C. `git checkout main; git reset main~1`**
>
> D. `git branch -D main`

删除上次提交的commit在某分支上。

1. 切换到分支
2. `git reset main~1` → reset 的功能是'撤销commit' ~几说明往回返几个

**Q5**. When running a command into their favourite pager `more` (so they can scroll it on the terminal) Alice notices the following unusual output. What is going on, and what are the weird `ESC[01;34m` bits?

```
$ ls --color=always | more
ESC[0mESC[01;34mBackupsESC[0m
ESC[01;34mDesktopESC[0m
ESC[01;34mDocumentsESC[0m
ESC[01;36mDownloadsESC[0m
ESC[01;34mMailESC[0m
ESC[01;34mMusicESC[0m
ESC[01;36mNotesESC[0m
ESC[01;34mPicturesESC[0m
ESC[01;34mReposESC[0m
ESC[01;34mVideosESC[0m
mbox
- (END)
```

A. They are part of the filenames

**B. They are ANSI color escape codes. The pager `more` doesn't know to translate them into colors**

C. The terminal has glitched and is displaying random noise

D. They are escape codes embedded as part of the filenames to make some filenames italic

不知道为啥，不过因为数字很小 → 推测是ANSI码

**Q7**. What is an *inode*?

A. An inode is an Apple music playing device

**B. An inode is the underlying datastructure used to store files on a disk**

C. An inode is a system call used to access files on a filesystem

D. In ODE means that the solution to this problem can be expressed as a series of ordinary differential equations

熟练运用man手册来做这种题。 **inode是用于在磁盘上存储文件的基础数据结构**

**Q19**. Which of the below would redirect both standard output and standard error from `myprogram` to `log.txt`?

A. `./myprogram > log.txt 1> log.txt`

**B. `./myprogram 1> log.txt 2>&1`**

C. `./myprogram 2>&1 > log.txt`

D. `./myprogram 2&1> log.txt`

**Solution:** A doesn't redirect stderr, C has the right idea but sends stderr to stdout before redirecting stdout (we covered exactly this issue in the lecture), D looks sort of plausible but simply isn't valid.

要是使用2>&1的话，txt文件应该在这个玩意儿前面。

这个命令执行了一个名为 `myprogram` 的程序，并将其标准输出（stdout）重定向到名为 `log.txt` 的文件中。而 `2>&1` 则是将标准错误（stderr）重定向到标准输出，意味着所有的输出（包括标准输出和标准错误）都会写入到 `log.txt` 文件中。

【2>&1是用来写标准错误的吗？

【是的，`2>&1` 是将标准错误（stderr）重定向到标准输出（stdout）的操作符。错误（stderr）重定向到文件的操作符。例如，`2> error.txt` 将标准错误输出到名为 `error.txt` 的文件中。

【单独2呢？

【`2>` 是将标准错误（stderr）重定向到文件的操作符。例如，`2> error.txt` 将标准错误输出到名为 `error.txt` 的文件中。

**Q20**. If `command2` ignores standard input and will only work with a file passed as an argument, how can we redirect the output of `command1` to be used by `command2`?

A. `command1 > command2`

B. `command2 $(command1)`

C. `command1 | command2`

**D. `command2 <(command1)`**

**Solution:** Note that we specified that command2 ignores standard input, so C would not work. B uses the output as an argument, which is not what command2 would understand, and A would just write command1 output to a file called command2.

Only work with a **file** passed → 排除C（标准流） B（arguments）

考察`<(`的作用 → 创建一个文件，所以有效

# Main Test:

**Q1**. Alice wishes to merge the branch `hotfix` into the branch `testing`. Which commands should they run?

**A.** `git checkout testing; git merge hotfix`

B. `git checkout hotfix; git merge testing`

C. `git checkout testing; git rebase hotfix`

D. `git checkout hotfix; git merge testing --no-ff`

考点：merge会合并branch，首先checkuot到 testing,在merge你要合并的哪个hotfix

**Q2**. Which option most accurately describes what the `git clone` command does?

A. `git clone` extracts a version of a file under version control at a specific commit point in history.

**B.** `git clone` **makes a local copy of an entire repository**

C. `git clone` fetches a file from Github and downloads it locally

D. `git clone` duplicates a repository on a remote Git forge (like Github)

在vagrant里调用手册 man git 或者 git 【特殊command】 - - help，`git clone --help`

**Q3**. When trying to fetch from a git remote Alice encounters the following error:

```
$ git fetch
ssh: Could not resolve hostname github.com: no address associated with name
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

What is the issue?

A. Their public key has not been correctly set up on Github.

B. The repository doesn't exist; they need to create it before they can pull;

C. There is a merge conflict

**D. Their network is down or misconfigured.**

看到ssh那里的hostname连接不上github问题，考虑是不是network问题。

**Q6**. Bob has the following id:

```
$ id
uid=1000(bob) gid=800(users) groups=800(users)
```

The passwd program has the following permissions:

```
$ ls -l `command -v passwd`
-r-sr-sr-x 1 root bin 21112 Feb 12 00:27 /usr/bin/passwd
```

What user and group will it run as?

**A. uid=root, gid=bin**

B. uid=root, gid=users

C. uid=bob, gid=users

D. uid=bob, gid=bin

首先看id，发现bob对文件来说属于other组别的。

其次看文件权限，发现 r-s是owner，r-s是group，r-x是other。

由于owner和group都被设置了s位（暂时以文件本身owner（uid）和本身group（gid）），所以就算bob运行这个程序，uid和gid由于被设置过，所以是root和bin。

**Q8**. What flag is needed to make the `ls` command list inode numbers?

**A. `-i`**

B. `-l`

C. `-n`

D. `-z`

熟练运用man手册。`man ls` 之后`/inode` 来查找是哪个number

**Q9**. Alice has the following Makefile. Since the last build, they have made some updates to `main.c`. Which rules will be run when next typing `make`?

```
main: main.c interface.o analysis.o

docs.html: main.c interface.c analysis.c library.c
	documentation-generator main.c interface.c analysis.c library.c -o $@

interface.o: interface.c config.c
analysis.o: analysis.c library.o
library.o: library.c
```

A. main

B. main interface.o analysis.o

**C. main docs.html**

D. main interface.o analysis.o docs.html

make已经编译后，修改文件之后的编译。方法是查看**修改**的文件在**哪几个命令里有**

这道题里，修改的main.c出现在main命令里和docs.html命令里。so...

What has gone wrong?

A. The Java compiler version is incorrect

B. The `pom.xml` is syntactically incorrect

C. The syntax of the `mvn` comand is wrong

**D. A dependency is missing**

第10题，考了building JAVA,可以考试的时候翻exercise里java部分观看依赖，性质，插件等

**Q11**. Bob wants to store a table containing students answers in an SQL database. The table should have 3 attributes:

- `student` (an 8 digit number which references the `id` attribute of the `student` table)
- `question` (a 2 digit number which references the `id` attribute of the `question` table)
- `answer` (text containing their answer)

What SQL command would create the most appropriate table?

A. `CREATE TABLE answers(student VARCHAR(8), question INTEGER, answer TEXT, FOREIGN KEY student REFERENCES student(id), FOREIGN KEY questic REFERENCES question(id), PRIMARY KEY (answer));`

B. `CREATE TABLE answers(student VARCHAR(8), question INTEGER, answer TEXT UNIQUE, FOREIGN KEY student REFERENCES student(id), FOREIGN KEY question REFERENCES question(id), PRIMARY KEY (student, question));`

C. `INSERT INTO answers(student, question, answer) VALUES ("0607970", 10, "C");`

**D.** `CREATE TABLE answers(student VARCHAR(8), question INTEGER, answer TEXT, FOREIGN KEY student REFERENCES student(id), FOREIGN KEY quest REFERENCES question(id), PRIMARY KEY (student, question));`

sql创建数据库。复合主键是(student, question). 主键是NOT NULL 且UNIQUE 的，answer可以有多个，所以不能设置UNIQUE性质

**Q12**. Bob wants to find each students mark in the exam. If each question is worth 1 mark, which SQL query will find the students' total scores?

You can assume the following tables in Bob's database:

Exam:

| Student | Question | Answer |
|---|---|---|
| 1 | 1 | A |
| 1 | 2 | B |
| 2 | 1 | A |
| 2 | 2 | C |
| 3 | 1 | - |
| 3 | 2 | - |

Answers:

| Question | Answer |
|---|---|
| 1 | A |
| 2 | B |

A. `SELECT student, COUNT(exam.answer = answers.answer) AS mark FROM exam JOIN answers WHERE exam.question=answers.question GROUP BY student_id;`

**B.** `SELECT student, SUM(exam.answer = answers.answer) AS mark FROM exam JOIN answers WHERE exam.question=answers.question GROUP BY student_id;`

C. `SELECT student, SUM(exam.answer = answers.answer) AS mark FROM exam JOIN answers WHERE exam.question=answers.question GROUP BY exam.question;`

D. `SELECT student, COUNT(exam.answer = answers.answer) AS mark FROM exam JOIN answers WHERE exam.question=answers.question GROUP BY exam.question;`

首先total分肯定排除了通过exam.qestion分组的。然后COUNT里不可以有表达式，但SUM里可以有。所以是B。（sql 各种小函数的运用）

**Q13**. The `FOREIGN KEY` constraint implies which other constraints (if any)?

A. `NOT NULL, UNIQUE`

B. `NOT NULL`

C. `UNIQUE`

**D. No other constraints**

外键没任何约束 → 对比主键，主键的约束是非空且唯一。

**Q14**. Bob is trying to run a command but it isn't working. To debug their issue they run their command with `strace`. What command are they trying to run?

```
execve("/usr/bin/cd", ["cd", "/home/jhl8636"], 0x7fff0476de18 /* 28 vars */) = 0
brk(NULL)                               = 0x1020000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=43207, ...}) = 0
```

这个题我都不知道要怎么继续考。 哪里看command是从第一行 ["cd", "/home/jhl8636"]里面

但是可以看出来他想run一个cd来着

**Q15**. Bob is trying to run a command but it still isn't working. Using the trace from before: what type of program does `strace` try and run, in the first instance?

A. A compiled binary command

**B. A shell script**

C. A shell builtin command

D. Impossible to say

-不能理解继续，助教的说法是：因为前面的操作很多所以

**Q16**. Bob is trying to run a command but it still isn't working. Using the trace from before: what's the problem?

A. The command they've tried to run doesn't exist

B. The permissions are set to prevent their command running

C. The filesystem and it's inodes are corrupt.

**D. A folder they've tried to access doesn't exist**

上面的strcae里有一句：不能写因为找不到文件。

```
write(2, "/usr/bin/cd: line 2: cd: /home/j"..., 66/usr/bin/cd: line 2: cd:
    /home/jhl8636: No such file or directory
) = 66
read(255, "", 26)                       = 0
```

**Q17**. Which of the following bash scripts would produce the output string "the easiest question"?

A.
```
#!/bin/bash
input="the hardest question"
echo input | sed s/easiest/hardest/
```

**B.**
```
#!/bin/bash
input="the hardest question"
echo $input | sed s/hard/easi/
```

C.
```
#!/bin/bash
input=the hardest question
echo "$input | sed s/hardest/easiest/"
```

D.
```
!#/bin/bash
input="the hardest question"
echo "input | sed s/easi/hard/"
```

**Solution:** There are clear mistakes in every other answer. A echoes 'input' rather than '$input' and gets the order of the sed s-expr components wrong. C assigns the string without quotemarks and echoes a string containing the sed command rather than piping input to that command. D also makes this mistake, gets the sed order wrong, and gets the shebang syntax wrong.

1.引用argument要加$,所以排除A，D。

2.input里搞的string但是没加""。

**Q18**. Alice is trying to set up key-based login on a service she currently accesses via using `ssh` and entering her password. Here are listings of her `.ssh` directory on her local machine:

```
-rw------- 1 alice alice 389 Jan 19 10:56 authorized_keys
-rw-r--r-- 1 alice alice 395 Feb 21 13:03 id_rsa.pub
-rw-r--r-- 1 alice alice 225 Feb 21 13:01 known_hosts
```

And on the server:

```
-r-------- 1 alice alice 395 Feb 21 13:30 authorized_keys
-rw-r--r-- 1 alice alice 395 Feb 21 13:21 id_rsa.pub
-rw-r--r-- 1 alice alice 225 Jan 19 11:02 known_hosts
```

There's a problem evident with her current setup. Identify which file's presence, absence or visible details indicates the problem.

A. The problem lies with 'authorized_keys' on the local machine.

B. The problem lies with 'known_hosts' on the server.

**C. The problem lies with 'id_rsa' on the local machine.**

D. The problem lies with 'id_rsa.pub' on the server.

考察了密钥和私钥。在ssh里的时候。 密钥只存在于local机器，公钥（有.pub后缀的）存在于server上。

SSH **严格控制权限的文件**主要包括：

1. **私钥文件（例如：id_rsa）**：私钥文件包含了与公钥配对的私钥，用于身份验证。通常应将其权限设置为 `600`，即只有所有者可读写。
2. **公钥文件（例如：id_rsa.pub）**：公钥文件包含了用于身份验证的公钥。虽然公钥文件的权限不太敏感，但通常建议将其权限设置为 `644`，即所有者可读写，其他用户只能读取。
3. **known_hosts 文件**：`known_hosts` 文件存储了已知主机的公钥。这个文件的权限应该设置得很严格，通常为 `600`，以防止其被非授权用户读取或修改。
4. **authorized_keys 文件**：`authorized_keys` 文件包含了用户允许登录到自己帐户的远程主机的公钥。这个文件的权限应该设置为 `600` 或 `644`，只允许所有者读写，并且在某些情况下，也许需要更严格的权限。

**Q21**. Look at the below shell session:

```
$ cp * ~/backup/
cp: cannot stat '*': No such file or directory
```

What can be said about the state of the filesystem and shell environment?

A. There is no file called '*' in the current working directory.

B. There is no file called 'cp' in the current working directory.

C. The current working directory is empty.

**D. All of the above.**

英国人可少玩点文字游戏。

# RESIT

**Q1**. Alice wishes to rebase the `bugfix` bugfix branch so that instead of being based on the `release-1.0` branch, it is based on the `release-1.1` branch. Which commands should they run?

**A. `git checkout bugfix; git rebase release-1.1`**

B. `git checkout bugfix; git merge release-1.1`

C. `git checkout release-1.1; git rebase bugfix`

D. `git checkout checkout release-1.1; git pull bug-fix`

考察rebase。 checkout到别的分支， rebase目标分支。

Q2原题不写了：`git clone` makes a local copy of an entire repository。

**Q3**. Bob maintains a fork of a popular application. After pulling from the main branch of the upstream repo Git reports an issue. When Bob runs `git status` it reports:

```
   git status
On branch openbsd
Your branch is up to date with 'github/openbsd'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
       modified:  .github/workflows/appcast.yml
       modified:  .github/workflows/lint.yml
       modified:  .github/workflows/publish.yml
       modified:  .github/workflows/test.yml
       modified:  CHANGELOG.md

Unmerged paths:
  (use "git add <file>..." to mark resolution)
       both modified: src/maestral/constants.py
```

What should Bob do next?

**A. Bob should check `src/maestral/constants.py` and see how the files have diverged**

B. Bob's merge has failed. He should abort with `git merge --abort`

C. Bob's rebase has failed. He should abort with `git rebase --abort`

D. Bob should check the `workflows` folder to see how the files have diverged.

产生了冲突，要进文件思考如何改进冲突。

**Q4**. Why might a developer want to use the *squash* feature when using `git rebase`

**A. To rework a series of commits into a set of coherent patches.**

B. To optimise the storage space a git repo uses.

C. To make single commit smaller.

D. To optimise the storage space a git remote uses to host a repository.

熟练运用`git rebase --help`手册来看 squash是干嘛的。

**Q5**. Bob has the following id:

```
$ id
uid=1000(bob) gid=800(users) groups=800(users)
```

The passwd program has the following permissions:

```
$ ls -l `command -v passwd`
-r-xr-sr-x 1 root bin 21112 Feb 12 00:27 /usr/bin/passwd
```

What user and group will it run as?

**A. uid=bob, gid=bin**

B. uid=root, gid=bin

C. uid=root, gid=users

D. uid=bob, gid=users

文件权限的考试题。注意UID/GID有没有被set。→ file permission → s

*非考部分 6-7

**Q6**. Alice is trying to create a link to a file. They try creating a *hard* link to the file and a *symbolic* link. They accidentally delete the original file, and are somewhat surprised to find that whilst the symbolic link is now broken, the hard link still works.

Why is the symbolic link broken?

**A. The original file's path is gone, so the symbolic link doesn't refer to a valid file on the path.**

B. The symbolic link would have been deleted when the original was deleted. Consequently the symbolic link no longer exists.

C. The file system does not support symbolic links: the link never worked.

D. The link does still work but the linked file is now empty, hence the confusion.

*[1 mark]*

**Q7**. Why does the hard link still work?

**A. The underlying inode is still there: deleting the original just decremented the reference count from 2 to 1.**

B. The file was copied when the hard link was made.

C. The file was copied to the hard link upon deletion.

D. The hard link seems to work now but its pointing to a free inode. The original data just hasn't been overwritten yet.

Q8. Alice is trying to build their code with a Makefile. The first time they built the code it
seemed to compile, but the second time it seemed to do nothing. Why?

A. The compiled code is up to date. No recompiling is required.

**Q9**. Alice has created a new program in Java using the Maven buildsystem.

- They build a template using `mvn archetype:generate`
- They edit their code by running `cd src/main/java/uk/ac/bristol/cs` and then `nano App.java`
- They run `mvn compile` to build their code

cd的路径就不对 so

**A. They are running the build from the `src` folder not the project root. Consequently, Maven cannot find the `pom.xml`.**

*[1 mark]*

**Q10**. Bob wants to store a table containing rankings of different flavours of soup. The table should have 3 attributes:

- `flavor` (text referencing `name` attribute of the flavour table)
- `ranking` (a number)
- `notes` (optional text containing notes)

What SQL command would create the most appropriate table?

**A.** `CREATE TABLE soupranking (flavor TEXT PRIMARY KEY, ranking INTEGER NOT NULL, notes TEXT, FOREIGN KEY flavor REFERENCES flavor(name));`

B. `CREATE TABLE soupranking (flavor TEXT, ranking INTEGER NOT NULL, notes TEXT, FOREIGN KEY flavor REFERENCES flavor(name));`

C. `CREATE TABLE soupranking (flavor TEXT, ranking INTEGER NOT NULL, notes TEXT, PRIMARY KEY (flavor, ranking), FOREIGN KEY flavor REFERENCES flavor(name));`

D. `CREATE TABLE soupranking (flavor TEXT NOT NULL, ranking INTEGER NOT NULL, notes TEXT, PRIMARY KEY (flavor, ranking), FOREIGN KEY flavor REFERENCES flavor(name));`

**Q11**. What normal form is Bob's soup ranking table in?

| Flavour | Ranking | Notes |
|---|---|---|
| Tomato | 8 | |
| Orange | 3 | Surprising and fruity |
| Chocolate Orange | 0 | Disgusting! |

**A. 3NF**

B. No normal form

C. 1NF

D. 2NF

用这张图举例范式，1NF满足，因为每列不可再分（不可能有新的竖列）

2NF满足，因为在这里Flavour是主键，完全被依赖

3NF满足，因为只能从Ranking本身，或者Notes本身推出Flavour，Ranking和Notes本身没互相推导的关系（这NF真的难懂

**Q12**. Bob wants to find what their mean soup ranking is.

Which SQL query will *not* tell them what they want?

**A.** `SELECT SUM(ranking)/COUNT(ranking) FROM soupranking;`

B. `SELECT AVG(ranking) FROM soupranking;`

C. `SELECT (1.0*SUM(ranking))/COUNT(ranking) FROM soupranking;`

D. `SELECT SUM(ranking)/(1.0*COUNT(ranking)) FROM soupranking;`

sql内置小函数的考察，本质其实是这种计算要得出正确答案只能用函数or转化浮点数

如何转化浮点数？ → 分子 or 分母 * 1.0

**Q13**. The `PRIMARY KEY` constraint implies which other constraints (if any)?

**A.** `NOT NULL, UNIQUE`

B. `NOT NULL`

C. `UNIQUE`

D. No other constraints

主键约束。 对应的非主键约束时没有约束。

**Q14**. Alice says Bob's code is dangerous. Bob says it isn't. The code doesn't seem to crash and compiles. Who is right and why?

```
final String query = "SELECT password FROM users WHERE user LIKE "+username;
```

**A. Alice is probably right: it looks like it may be vulnerable to SQL injection.**

B. Bob is probably right: it doesn't crash

C. Alice is probably right: there is a buffer overflow

D. Bob is probably right: they have used prepared statements

要在外面+ 双引号 → prepared statement. 不然就会SQL injection

**Q15**. Bob is trying to run a command but it isn't working. To debug their issue they run their command with `strace`. What command are they trying to run?

```
execve("/usr/bin/ls", ["ls", "/hoem"], 0x7ffef4661138 /* 45 vars */) = 0
brk(NULL)                               = 0x1ca1000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
    0x7fa9994c8000
```

看第一行 就是command ： `ls /hoem`

Q16. Bob is trying to run a command but it still isn't working. Using the trace above: what's
the problem?

A. They've made a typo in their file path

**Q17**. What is `ltrace` and what is it used for?

**A. It traces which library functions a program uses**

B. It is an interactive debugger that provides breakpoints and stepping functionality

C. It traces what system calls a program uses

D. It is the *Linux / Tux Race* game: a modern 3D games program for the Linux enthusiast

**Q18**. Alice is trying to set up key-based login on a service she currently accesses via using `ssh` and entering her password. Here are listings of her `.ssh` directory on her local machine:

```
-rw------- 1 alice alice 389 Jan 19 10:56 authorized_keys
-rw------- 1 alice alice 395 Feb 21 13:03 id_rsa
-rw-r--r-- 1 alice alice 395 Feb 21 13:03 id_rsa.pub
-rw-r--r-- 1 alice alice 225 Feb 21 13:01 known_hosts
```

And on the server:

```
-rw-r--r-- 1 alice alice 395 Feb 21 13:21 id_rsa.pub
-rw-r--r-- 1 alice alice 225 Jan 19 11:02 known_hosts
```

There's a problem evident with her current setup. Identify which file's presence, abscence or visible details indicates the problem.

A. The problem lies with 'authorized_keys' on the local machine.

**B. The problem lies with 'authorized_keys' on the server.**

C. The problem lies with 'known_hosts' on the local machine.

D. The problem lies with 'id_rsa.pub' on the server.

**Solution:** The absence of an authorized_keys file on the server is a clear problem for key-based login. The permissions on the files are all correct. Moreover, 'authorized_keys' serves no purpose locally, and 'known_hosts' and 'id_rsa.pub' don't matter on the server. We went through this key-based setup process in the SSH labs at the start of Part 1.

**Q19**. Which of the below would redirect both standard output and standard error from the command `noisy` to `/dev/null`?

A. `noisy > /dev/null 1> /dev/null`

B. `noisy 2&1> /dev/null`

**C. `noisy 1> /dev/null 2> /dev/null`**

D. `noisy 2>&1 > /dev/null`

**Solution:** A doesn't redirect stderr, D has the right idea but sends stderr to stdout before redirecting stdout (we covered exactly this issue in the lecture), B looks sort of plausible but simply isn't valid. C would have output problems if it went to a file, but since we're redirecting to /dev/null we clearly don't care about that.

考察shell的输出。 假如要用2>&1的话，D应该写成 `nosiy > dev/null 2>&1`

**Q20**. What would be the output when executing the following piped command?

```
echo "words in a string" | sed 's/in //' | head -1
```

**A. "words a string"**

B. "words in a"

C. "words"

D. "words in in a string"

**Solution:** We covered both sed and head in the piping exercises in the POSIX labs. The sed command removes one of the words in the original text, but head would have no effect on this output because it operates on lines.

放到shell里就行..... 这个command的意思是把'in '转换成无。

**Q21**. The Unix philosophy suggests that:

A. Programs are imitations of abstract eternal ideals.

B. Programs do not need to be able to cooperate.

C. The universal interface between programs should be IP packets.

**D. None of the above.**

**Solution:** This should be straightforward. B and C are pretty obviously corrupt, A would be some sort of programmer-Platonism. We covered the Unix philosophy in the Pipes 1 lecture.

unix philosophy 强调： 1.小而美2.合作完成目标3.在接口的东西是TEXT STREAM

**Q22**. If `command2` needs to be given a string argument that comes from the output of `command1`, how can we capture the output of `command1` to be used by `command2`?

A. `command1 > command2`

**B. `command2 $(command1)`**

C. `command1 | command2`

D. `command2 <(command1)`

**Solution:** Note that we specified that command2 requires an argument, so A and C are misguided. D uses the output as contents of a dummy file, which is not what command2 would understand. This redirection is covered in the Pipes lecture.

$(command1) 表示argument