

CS246 Course Notes

mzx!

July 9, 2019

Chapter 1

Midterm Structure

1.1 breakdown

7-8 Questions in Total

1.2 Definition Questions

Q.1 Multiple Choice

Note that sometimes there are more than one correct choices

Q.2 True or False

Q.3 Short Answers

1.3 Long Answer Questions

Q.4 Bash Programming

Q.5 Big 5 Programming

Q.6 Big 5 Programming

Q.7 Finding Error

1.4 NOT IN TEST

Make File

static

Chapter 2

Linux Shell Bash

2.1 Relative Path & Absolute Path

Absolute Path always starts with `/` or `~/`, and starts at root directory

Relative Path starts from current directory, such as `bin/bash`

2.2 Output Redirection & Input Redirection

Output Redirection executes commands args and captures the output in the file instead of screen

Input Redirection take the input from the file instead of the key board and display

2.2.1 Shell Command About Directory

For Relative Path we use `.` as Current Directory (Relative), `..` as Parent Directory, `../..` as GrandParent (Relative), Note that it is not "...".

For Absolute Path we use `~` as Home Directory and `~userid` as User's Home Directory (Requires Permission)

2.2.2 stdin/stdout/stderr

`0` stdin, using `<` for input redirection

`1` stdout, using `>` for overwrite output redirection, using `>>` for append output redirection

`2` stderr

2.3 Cat

```
1 cat input.txt // cat file
2 cat < input.txt // cat stdin
```

Both command gives same output

2.4 Word Count

Output format: `newline words bitscount`

```
1 //in.txt:
2 //this is it
3
4 wc in.txt // wc file
5 wc < input.txt //wc input
```

These two command gives **DIFFERENT** output, where "wc file" gives `1 3 3 in.txt` but "wc input" gives `1 3 3`. Note that the file name will not be produced with "wc input"

2.5 Pattern Matching

2.5.1 WildCard Mathing/Globbering Matching

WildCard Matching uses `*` as "0 or more any chars", `?` as single any char.

```
1 *.cc // Any .cc files
2 test.* // Any file with name "test" no matter the extend
3
4 ?.cc //all .cc file with a name in one char
```

2.5.2 Regular Expression

In Regular Expression `.` represent **ANY SINGLE CHAR**. `+`: 1 or more preceding pattern. `*`: 0 or more preceding pattern. `ibx?`: 0 or 1 preceding pattern

Note that `*` in WildCard Matching = `.*` in Regular Expression

`^` :Beginning of a line, `$` : End of a line.

`(...—...—...)` : any one the given expressions

`[...]` : any character form this set `[^...]` or `[!...]` any one of the characters not listed

2.6 Quoting

`\` : Escapes any character

`"` : Open a shell and run what's in it in background

`"` : Don't allow variable to be expand, literally whatever is inside

`""` : Allow variable to expand, recognize escapes, back quotes, but DO NOT RUN COMMAND

2.7 Pipe

Pipes use the output of one program as the input of another

2.8 Testing

Testing cannot guarantee your program is correct, but can prove it's incorrect

Chapter 3

Shell Scripts

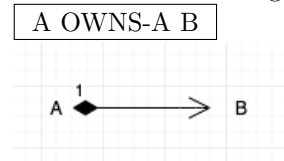
Chapter 4

Inheritance

4.1 UML

4.1.1 OWNS-A / Composition

when an object takes ownership of another object, Composition create an Owns-A relation. Embedding an object within another is composition



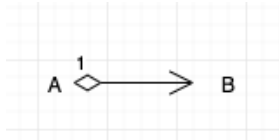
- If A is copied, B is copied as well
- If A is destroyed, B is destroyed

Examples Basis OWNS-A Vector, Car OWNS-A Wheels

```
1 class Car{
2     Wheel w1,w2,w3,w4;
3     // With automatic delete w1,w2,w3,w4
4     // If I want to destroy a car, its wheels get
5     // destroyed as well
6     // If I want to copy a car, the duplicated car has
7     // wheels w1 w2 w3 w4 as well
8 }
```

4.1.2 HAS-A / Aggregation

A HAS-A B



- When A is copied, B is NOT
- When A is destroyed, B is NOT
- NOTE that someone else should take the ownership of B and delete B at some point

Examples Lake HAS-A Geese

```
1 class Lake {
2     Goose **geese;
3 };
4 //If i want to delete the Lake, we are not gonna
5     delete geese, but someone else should
    //We will delete the array, that contain the geese
```

4.1.3 IS-A / Inheritance

Inherits ALL(Public and Private) members(fields as methods) from the Base Class



Book Collection

- Text IS-A Book (with an extra topic field)
- Comic IS-A Book (with an extra hero field)
- Any method you can call on Book Object, you can call on a Text Object
- Text inherits private fields, title, authors, numPages from BaseClass Book, they are not declared in Text class
- Yet Text objects CANNOT access those fields
- Yet we can not use MIL to construct a Text class