

A teal-colored 3D pyramid is centered on the slide. The background is a grayscale aerial view of a city with many skyscrapers. The pyramid has a semi-transparent effect, showing the cityscape through its faces.

程设实验Project02

Linux下类Shell实现

191220090 沈天杰

myShell

1. 需求分析
2. 数据结构
3. 模块划分
4. 模块关系
5. 核心算法
6. 实验过程
7. 拓展功能



/ 01 需求分析



类Shell实现

- 1. 实现自己的shell
- 2. 命令有：
cp (-r) file1 file2
cmp file1 file2
wc -c/-w/-l file
cat file
man
sh
cd
- 3. 命令的格式与功能与Linux下的一致（参数尽量完整实现）
- 4. 一些附加功能

类Shell命令

`cp SOURCE DEST`

可以将SOURCE文件复制为DEST文件；如果DEST是一个存在的目录，则将SOURCE文件复制到该目录下。

`cp -r SOURCE DEST`

复制SOURCE目录下的所有内容到DEST

`cmp FILE1 FILE2`

比较FILE1和FILE2

`wc (-c/-w/-l) FILE1`

统计FILE1中的字节数/字数/行数

`cat FILE1`

显示FILE1中的内容

`man COMMAND`

显示COMMAND的说明文档

`sh test.sh`

执行脚本文件中的大量Shell命令

`cd`

目录切换

/ 02 数据结构



数据结构

需要处理哪些信息？

文件/文件夹

```
1 #ifndef COMMAND_H
2 #define COMMAND_H
3 #include<string>
4
5 //欢迎界面
6 void welcome_menu();
7 bool get_command(const string str,const string s);
8 //具体功能
9 int copy_file(const string file1,string file2);
10 void copy_all(const string str1,string str2);
11 void search(const string str1,const string str2);
12 void copy_dir(const string dest_path);
13 int cmp(const string file1,const string file2);
14 void wc(const string option,const string file);
15 void cat(const string file);
16 void man(const string option);
17 void print_man(const string file);
18 int sh(const string file);
19 void my_cd(string path);
20 #endif
```

文件读写操作
比较简单

随建随用
比较灵活

未选择专门存储

1 运用iterator 复制普通文件

```
in_file>>noskipws;//读取输入流的空白字符
istream_iterator<char> is(in_file);
istream_iterator<char> eof;
ostream_iterator<char> os(out_file);
copy(is,eof,os);
in_file.close();
```

2 相关库函数 处理文件夹

```
DIR *src_dir=opendir(src_path);
DIR *dest_dir=opendir(dest_path);
struct dirent* entry=NULL;
struct stat state_of_entry;
char* curr_dir=getcwd(NULL,0);
while((entry=readdir(src_dir))!=NULL)
```

3 字符串一行一行 来处理文件信息

/ 03

模块划分



模块划分

h

1. command. h

cpp

1. main. cpp

2. menu. cpp

3. function

func_cp. cpp
func_cmp. cpp
func_wc. cpp
func_cat. cpp
func_man. cpp
func_more. cpp

欢迎与提示界面

调用

运行命令

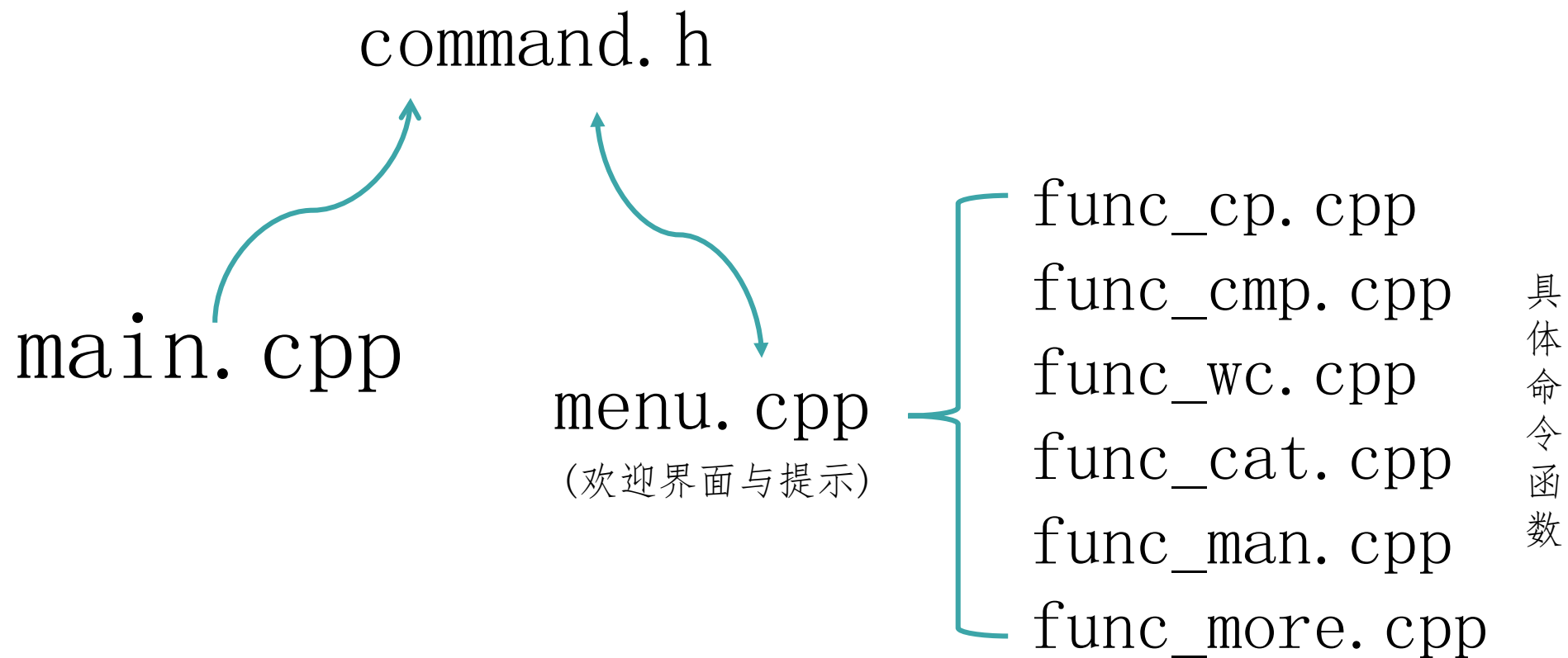


/ 04

模块关系



模块关系



/ 05 核心算法



深度优先遍历 (DFS)

算法描述：

沿着文件夹遍历文件，尽可能深的搜索文件夹的内容，当某一个文件夹 v 都已经搜寻到文件时，搜索将回溯到发现文件夹 v 的最初文件夹。整个进程反复进行直到所有文件都被访问为止。

伪代码：

- 1、如果是文本文件，到达遍历边界
- 2、如果是文件夹
 - 继续下一步
 - 回溯初始状态

/ 06 实验过程



具体实现

- 一致路径：while循环中输入命令进行匹配执行相应函数
- cp: 用iterator的copy函数复制文本文件，递归复制目录下内容。（先判断相应目录是否存在，若不存在用mkdir创建新的目录）
- cmp: 用getline一个回车符一个回车符读取文本文件存储到string变量中进行比较
- cat: 用getline一个回车符一个回车符读取文本文件存储到string变量中输出

具体实现

- wc: 用getline一个回车符一个回车符读取文本文件存储到string变量中分别进行统计。
(统计方法: l用\n数, c用sizeof, w用空格, 跳格和换行符)
- man: 存储好每个文件的说明放在同级目录的man文件夹中, 使用时对相应文件进行cat操作。
- sh: 正确读取参数, 一一执行命令

- 命令的参数数量不一

用getline以\n为分割存储到string变量中，再用空格进行分割放到vector<string>中，根据其size()判断具体执行命令

- man的切换屏幕

由ncurses库帮助

/ 07 拓展功能



拓展功能

- 执行脚本文件，Linux的Shell可以创建脚本文件(.sh)，执行脚本文件中的大量Shell命令，你的Shell应该有相应的功能。
- cp 在文件和文件夹间复制
- 路径切换cd
- Man切换屏幕输出（终端大小够大）
- 模仿shell命令形式 主机名+当前路径

An aerial photograph of a dense city skyline, likely New York City, featuring numerous skyscrapers and buildings. The image is partially obscured by a solid teal-colored horizontal band across the middle. The word "Thanks" is centered within this band in a black, serif font.

Thanks