TINY SHELL 模块4

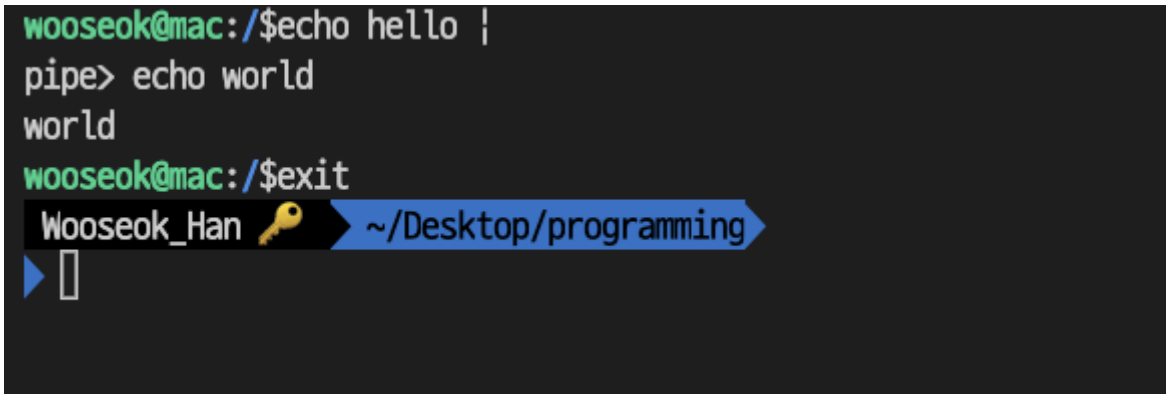
韩佑硕 2021080070 软12

基本功能完成情况

尽量按照真实的情况，完成了解释器框架，echo指令和复合指令的模拟实现。

附加功能 ： 复合指令的扩展语法和exit指令

1. 复合指令扩展语法 ： 当用户输入`"[指令 1] |"` 时， （也可以出现多条指令，重要的是最后以`"|"`结束）解释器输出`"pipe>"`等待用户再输入下一个指令，用户输入后，处理那条指令
2. `exit指令` ： 当用户输入`"exit"`时， 释放所有已申请的动态分配，并结束程序



源代码说明

1）main

类型 ： 返回 `int`

功能 ： 一开始设置计算机名，根目录， 和用户名，并实现接受指令的框架。

> 1. 生成提示符（`getPrompt`）
> 2. 如果用户只输入回车或空格，则重新输入（`checking_valid`）
> 3. 产生标准输入的一个复制本（command），用于拆token（在`splitToken`实现)
> 4. 用`splitToken`函数在`argv`储存各个token之后，用`get_strout`函数输出最终的标准输出（其中可能会输出异常指令）
> 5. 释放动态分配的内存（`delete_memory`）
> 6. 最终释放内存并如果没有exit指令（`left_argc !=0`），用户再次输入指令，如果有（`left_argc == 0`）， 则结束程序

用复制本（`command`）的原因 ： ```splitToken``函数会改变作参数的字符串，而有些指令需要从标准输入读取，为了保护标准输入，用复制本

2）getPrompt

类型 ：返回 `char**`

参数 ：`(char **)` 实际上传的是（`prompt`）

功能 ： 生成及更新提示符。

为了执行cd指令改变工作目录之后方便地更新提示符，这个函数每次输出提示符之前被运行

3) checking_valid

类型 ： 返回 `bool`

参数 ：`( int , char** )` 实际上传的是 `( cmdSize , prompt )`

功能 ： 判断用户输入的命令是不是只含有回车，或空格。如果命令无效（只含有回车空格）则释放prompt储存的内存，并返回FALSE，否则（有效）返回 TRUE

4) splitToken

类型 ：返回 `int`

参数 ：`(char* , char* [] , int)` 实际上传的是 `( command , argv , cmdSize )`

功能 : 把标准输入里的字符串拆成若干个token(用 `getToken`)，存储到`argv`，并返回总token个数（后作`delete_memory`函数的参数）

5 ）getToken

类型 ：返回 `char*`

参数 ：`( char * )` 实际上传的是 *command*的复制本 `( commandcp )`

功能 : 用`strtok`函数拆开标准输入的复制本`(command)`，分解符为 `" |"` ，返回拆开的一个token

是splitToken函数里调用的函数，为了节省内存，每次动态分配char*型数组，后来用 delete_memory函数释放

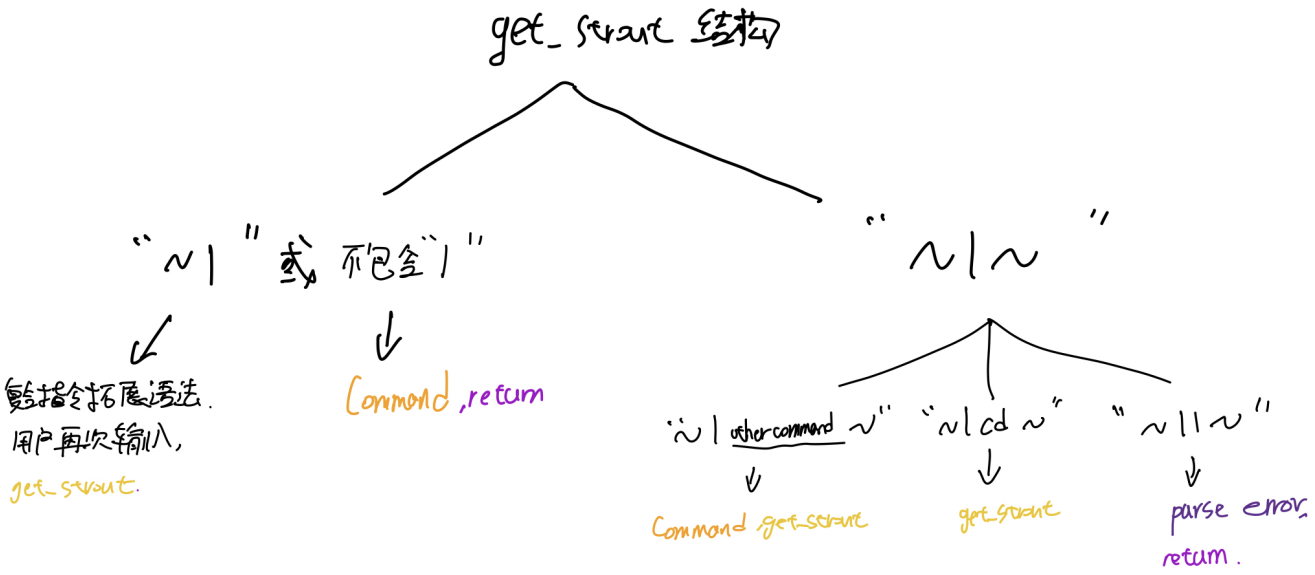
6 ）get_strout

类型 ：返回 `int (left_argc)`

参数 ：`( int , char* [])` 实际上传的是 `( argc , argv )`

功能 : 实现复合指令，得最终标准输出，并输出到屏幕上

> 1. 递归函数，执行到`argv`不再包含 `"|"`为止
> 2. 用 `if , else` 语句和 `switch`语句 分类讨论
> 3. 主要分成 1 ）“~ |”或者 "不包含 |" 和 2 ）“~ | ~”的情况
> 4. 执行作业里的功能（通过`Command`）及`exit`功能（直接实现）
> 5. 实现复合指令的扩展语法（`~|`的形式出现时）



反映了在真实情况下，cd指令不是最后一个指令的话，不被执行的情况和多条复合指令多次出现的话（ex)| |），输出“parse error near '|'”的情况 （这里没考虑 double vertical bar ||，也当复合指令重复出现处理）

6）Command

类型 ：返回 int

参数 ：（int， char *[]）实际上传的是（argc，argv）

功能：判别argv[0]，而判断要执行什么指令，如果```argv[0]``的指令是已知的（作业里的，或拓展功能），则返回0，否则（不存在->第一类异常指令）返回1，特别，是复合指令的话，返回2

> 1. 用 if，else 语句
> 2. 这函数的返回值作get_strout函数里switch语句的参数（switcher）

7）delete_memory

类型 ：void

参数 ：（char **， int， char *， cahr *）实际上传的是（prompt，argc，argv，command）

功能：释放所有在main函数申请的动态分配（包括getPrompt，splitToken函数和command），并为了下一次指令的输入，清空标准输入

8）doDiff

类型 ：void

参数 ：（int， char *）实际上传的是（argc，argv）

功能：从argv[0] 到 argv[argc]的字符串拷贝到标准输出

> 1. 不读取标准输入

> 2. 用`if，else`语句实现
> 3. 选`--help`选项时，从`"echo--help.txt"`文件读取内容，拷贝到标准输出

测试文件及测试代码说明

为了有效地解释本人所写的代码能做到的功能，写了测试文件和测试代码

测试文件里的内容由用户能输入的若干命令组成，比如，`echo hello | world | echo world`

每行是一条指令，测试代码从测试文件读取每条指令，并在屏幕上输出其结果

只是为了实时看到所输入指令的结果，在测试代码中引入了 `unistd.h`里的 `sleep( )`函数， (`cmd.cpp`代码不包含`unistd.h`)