

CS205 C/ C++ Programming Assignment 2

Name: 傅伟堡(Weibao Fu)

SID:11812202

Part 1-Analysis

The problem is to simulate a terminal. We are asked to write a program that can follow some commands. We also need to use "switch" statement instead of using "if...else" structure. By the way, to ensure my program will not crash, I also need to consider the bad input. In the terminal, the program will ignore the letter caes, so my program should also recognize the commands ignoring the case.

Part 2-Code

```
#include<stdio.h>
#include<iostream>
#include<cstring>
#define Maxn 1000
using namespace std;
char str[Maxn];
const char *commands[5]={"start","stop","restart","reload","exit"};

void deleteblank(){
    int l=0,r=strlen(str)-1,id=0;
    while(str[l++]==' ');
    while(str[r--]==' ');
    for(int i=l-1;i<=r+1;i++){
        if(str[i]<='Z'&&str[i]>='A') str[i]+='a'-'A';
        str[id++]=str[i];
    }
    str[id]='\0';
}

int main(){
    while(true){
        int cmd=-1;
        printf("> ");
        cin.getline(str,Maxn);
        deleteblank();
        if(str[0]=='\0') continue;
```

```

    for(int i=0;i<5;i++){
        if(strcmp(str,commands[i])==0){
            cmd=i;
            break;
        }
    }
    switch(cmd){
        case 0: printf("command <start> recognized.\n");
                break;
        case 1: printf("command <stop> recognized.\n");
                break;
        case 2: printf("command <restart> recognized.\n");
                break;
        case 3: printf("command <reload> recognized.\n");
                break;
        case 4: return 0;
        default: printf("Invalid command.\n");
    }
}
return 0;
}

```

Part 3-Result & Verification

Test case #1

Input: start
stop
restart
reload
exit

Output: command <start> recognized.
command <stop> recognized.
command <restart> recognized.
command <reload> recognized.

```

> start
command <start> recognized.
> stop
command <stop> recognized.
> restart
command <restart> recognized.
> reload
command <reload> recognized.
> exit

-----
Process exited after 16.33 seconds with return value 0
请按任意键继续. . .

```

Test case #2

Input:

```
start
        stop
            restart
wrong input
sstart
exit
```

Output: command <start> recognized.
command <stop> recognized.
command <restart> recognized.
Invalid command.
Invalid command.

```
>
>
> start
command <start> recognized.
>         stop
command <stop> recognized.
>         restart
command <restart> recognized.
>         wrong input
Invalid command.
>         sstart
Invalid command.
>         exit
-----
Process exited after 4.201 seconds with return value 0
请按任意键继续. . .
```

Test case #3

Input: SStart

```
        STOP
            rESTaRT
RREload
eXIT
```

Output: command <start> recognized.
command <stop> recognized.
command <restart> recognized.
Invalid command.

```

> SStart
command <start> recognized.
>                STOP
command <stop> recognized.
>                rEsTaRT
command <restart> recognized.
>                RRELoad
Invalid command.
>                eXIT

-----
Process exited after 2.487 seconds with return value 0
请按任意键继续. . .

```

Test case #4

```

Input:  %%%&&&
        sta rt
        restar_t
        relo-ad
        2222
        exit
Output: Invalid command.
        Invalid command.
        Invalid command.
        Invalid command.
        Invalid command.

```

```

> %%%&&&
Invalid command.
>        sta rt
Invalid command.
>        restar_t
Invalid command.
>        relo-ad
Invalid command.
>        2222
Invalid command.
>        exit

-----
Process exited after 4.84 seconds with return value 0
请按任意键继续. . .

```

Part 4-Difficulties & Solutions

1. Using `scanf()` cause we may only read the data before `' '`, and leave the other information in the cache. And it may cause my program crash. So

`cin.getline()` is used to read the whole line.

2. Since the bad input may have some blank before the commands or at the end of the commands, so we need to delete these blanks at the beginning.

3. Another problem we should notice is that if the user only input the blanks, the program should prompt the user to input the command again.

4. Actually, in the terminal the commands' execution will ignore the case.

So, we should also ignore the letter cases in my program. Such that, if the input include upper cases, the program should execute normally.