

咕 咕 咕？

zhx

今天的内容非常奇怪且杂乱

系统环境的配置

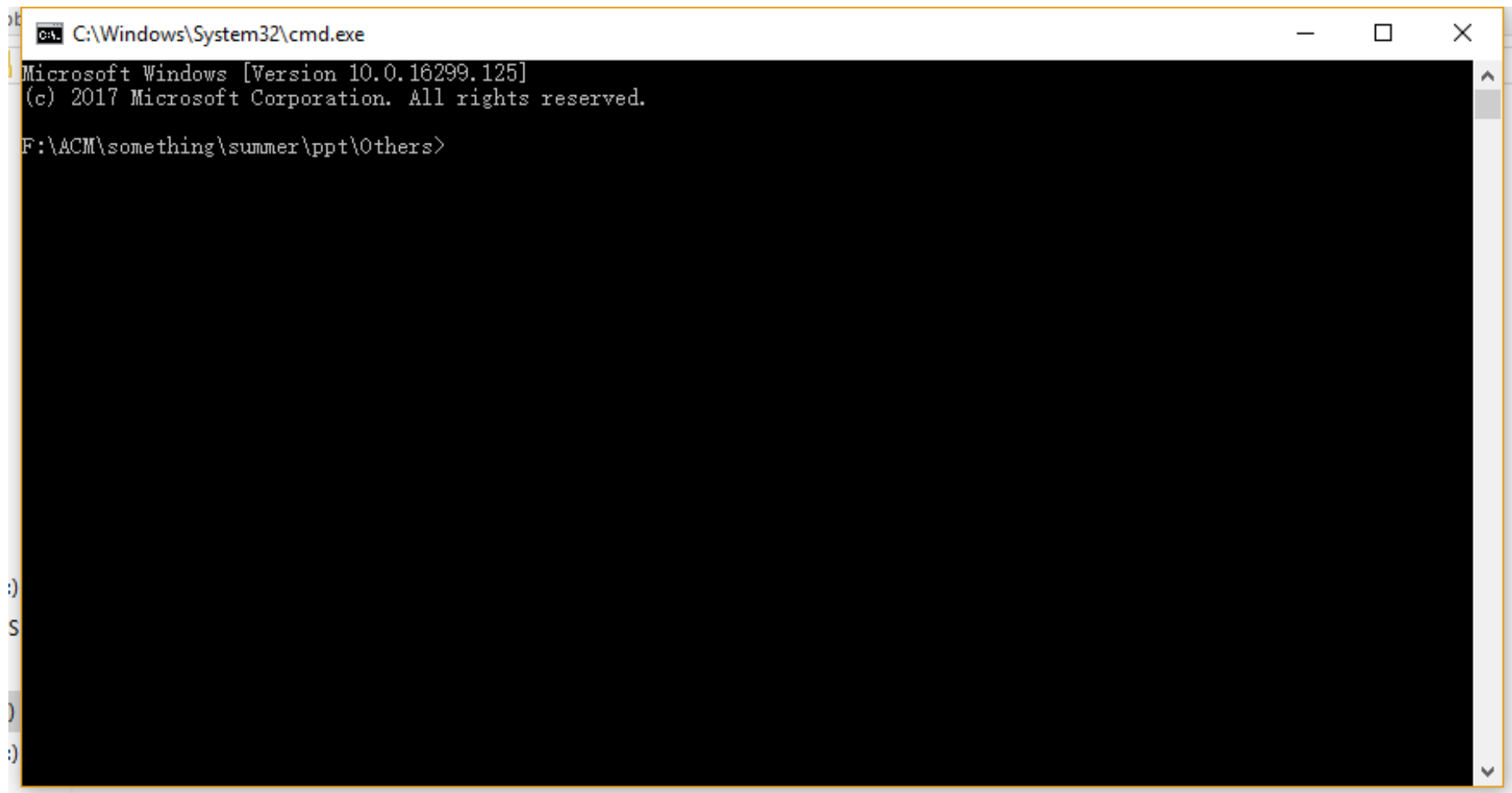
- ▶ Linux不需要配置
- ▶ 对于windows:
- ▶ 控制面板——系统——高级系统设置（不同版本可能不太一样）
- ▶ 环境变量——如果没有Path这一项 请手动添加
- ▶ 打开Path项 将dev或者mingw下bin文件夹的路径添加进去

启动命令行

- ▶ 对于windows7及以上的版本 直接在地 址栏输入cmd即可在对应文件夹启动命令行
- ▶ 对于任何windows版本 windows+R——输入cmd 可以在windows根目录启动命令行
- ▶ 对于windows 10玩家，请自行安装Ubuntu on windows的虚拟环境
- ▶ 对于linux玩家，ctrl+alt+t



命令行



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

F:\ACM\something\summer\ppt\Others>
```

命令行基本操作——windows

- ▶ 进入某一个文件夹 `cd 文件夹名字`
- ▶ 返回上一级 `cd ..`
- ▶ 去D盘 `D:`
- ▶ 显示文件夹下有啥东西 `dir`
- ▶ 运行程序 `a.exe`

- ▶ Tab is all you need

命令行基本操作——linux

- ▶ Linux的文件夹切换同windows
 - ▶ 只是没有盘的概念
 - ▶ 显示当前文件夹里面的东西 ls
 - ▶ 运行程序 ./a
-
- ▶ Tab is all you need

编译的基本操作

- ▶ Windows: `g++ a.cpp -o a.exe -g -Wall && size a.exe`
- ▶ Linux: `g++ a.cpp -o a -g -Wall && size a`
- ▶ `-o a.exe` 指定可执行文件为a.exe
- ▶ `-g` 可调试
- ▶ `-Wall` 显示所有warning
- ▶ `size a.exe` 测一下你程序所需要的内存
- ▶ `-O2`: `g++ a.cpp -o a.exe -O2`
- ▶ 在这个位置加就可以
- ▶ 请确保在同一目录下运行

```
C:\work\test>g++ a.cpp -o a.exe -g -Wall && size a.exe
   text    data     bss     dec     hex filename
  13156    1572     124    14852    3a04 a.exe
C:\work\test>
```

运行程序

- ▶ a.exe
- ▶ 重定向输入输出:
- ▶ a.exe < in.txt > out.txt

```
C:\work\test>a.exe
```

```
1
```

```
2
```

```
3
```

```
C:\work\test>a.exe < in.txt > out.txt
```

```
C:\work\test>
```

调试程序

- ▶ gdb a.exe 进入调试环境

```
C:\work\test>gdb a.exe
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\work\test\a.exe...done.
(gdb)
```

Gdb 基本操作

- ▶ `r` 运行程序
- ▶ `p x` 打印变量`x`的值
- ▶ `b 123` 在第123行设置断点
- ▶ `b gg` 在函数`gg`设置断点
- ▶ `s` 执行下一步
- ▶ `n` 执行下一行（即直接执行完该行）
- ▶ `display x` 将`x`加入常视变量列表
- ▶ `c` 执行到下一个断点
- ▶ `q` 退出gdb

gdb 基本操作

- ▶ `l x` 显示x行的代码
- ▶ `del x` 删除编号为x的断点
- ▶ `del` 删除所有断点
- ▶ `undisplay x` 不再显示x的值

Gdb 基本操作

- ▶ `call f(x)` 调用一次函数f，以x为参数
- ▶ `b 123 if (a == 3)` 条件断点，在a=3会停下来的断点
- ▶ `finish` 直接完成并退出当前函数

Gdb 基本操作

- ▶ 需要一段时间熟悉这个东西
- ▶ 熟悉了之后你会发现dev是什么辣鸡玩意儿

G++ 进阶操作

- ▶ `g++ a.cpp -o a.exe -g -Wl,--stack,41943040`
- ▶ 开大栈空间
- ▶ `g++ a.cpp -o a.exe -g -pg`
- ▶ `a.exe`
- ▶ `gprof a.exe > test.log`
- ▶ 新世界大门

命令行进阶操作

- ▶ Windows: `fc 1.out 2.out`
- ▶ Linux: `diff 1.out 2.out`

C++ 基本操作：读入优化

- ▶ `Int getint(){`
- ▶ `Char c = ' ';`
- ▶ `While (c<'0' || c>'9')`
- ▶ `c=getchar();`
- ▶ `Int ans=0;`
- ▶ `While (c>='0' && c<='9')`
- ▶ `{Ans=ans*10+c-'0';c=getchar();}`
- ▶ `Return ans;`
- ▶ `}`

C++ 基本操作：输出优化

- ▶ 将所有输出连接为一个字符串一起输出

C++ 基本操作：Windows 对拍

```
▶ Int main(){  
▶ While(true) {  
▶ System("data.exe");  
▶ System("std.exe");  
▶ System("brute.exe");  
▶ If (system("fc std.out brute.out")) break;  
▶ }  
▶ }
```

C++ 基本操作：Linux对拍

- ▶ `Int main(){`
- ▶ `While(true) {`
- ▶ `System("./data");`
- ▶ `System("./std");`
- ▶ `System("./brute");`
- ▶ `If (system("diff std.out brute.out")) break;`
- ▶ `}`
- ▶ `}`

C++ 基本操作：造数据

- ▶ Rand
- ▶ Windows: 0-32767
- ▶ Linux: 0-2147483647

C++ 基本操作：造数据

- ▶ 生成一个排列：`random_shuffle(z+1,z+n+1)`
- ▶ 生成一张图
- ▶ 生成一棵树
- ▶ 生成一个深度很大的扫帚树
- ▶ 生成一张章鱼图

C++ 基本操作：卡时

- ▶ `#include<ctime>`
- ▶ `Int t=clock();`
- ▶ `CLOCKS_PER_SEC`

C++ 基本操作

► `Int main(int argc,char *argv[])`

C++ 基本操作

- ▶ Double a,b,c;
- ▶ $a-b+c$
- ▶ $a+c-b$

C++ 基本操作

- ▶ `Int a[100];`
- ▶ `Int b[100];`
- ▶ `A[150] = 50;`

Vim 基本操作

- ▶ Vim的熟悉也需要时间
- ▶ 熟悉之后你就会觉得dev是什么辣鸡玩意儿

Vim 基本操作

- ▶ Windows: 双击图标打开, 或者右键cpp文件选择用vim编辑
- ▶ Linux: 在命令行直接输入vim即可

Vim 基本操作

- ▶ 进入之后默认为控制状态
- ▶ 在控制状态下按下i可以进入编辑模式
- ▶ 在编辑模式下可以正常写代码
- ▶ 编辑模式下的一个快捷键:
- ▶ Ctrl+p 自动补全
- ▶ 任何时候, 想要回到控制状态不断按 esc就行了

Vim 基本操作

- ▶ 在控制状态下的一些命令:
- ▶ :w 保存
- ▶ :w a.cpp 保存为a.cpp
- ▶ :q 退出当前窗口
- ▶ :wq 保存退出
- ▶ :e a.cpp 打开a.cpp
- ▶ :vsplit a.cpp 在左边打开a.cpp
- ▶ :split a.cpp 在上面打开a.cpp

Vim 基本操作

- ▶ `:123` 跳到123行
- ▶ `!:???` 在命令行中执行某条指令
- ▶ `:w!` 强制保存
- ▶ `:q!` 强制退出
- ▶ `colorscheme ????` 设置vim颜色风格

Vim 的 配 置

- ▶ 配置文件的打开方式:
- ▶ Windows 在上面菜单栏找到启动设定即可
- ▶ Linux 在控制模式下 :e .vimrc
- ▶ Windows 会有自带的配置文件，讲给定的配置文件写在前面即可

Vim 的 配 置

- ▶ `set number`
- ▶ `set tabstop=4`
- ▶ `set softtabstop=4`
- ▶ `set shiftwidth=4`
- ▶ `set mouse=a`
- ▶ `set smartindent`

- ▶ `syntax on`

- ▶ `colorscheme evening`

Vim 的 配 置

- ▶ 配置g++和gdb:
- ▶ map <F9> : !g++ % -o %< -g -Wall && size %< <CR>
- ▶ map <F5> : !gdb %< <CR>

Python 入门

- ▶ Python的基本东西还是和C++差不多的，只是语法上稍有区别
- ▶ $A=b+c$
- ▶ `For a in range(0,n):`
- ▶ `If a=b and c=d:`
- ▶ Python的begin是：
- ▶ 区分不同代码段用缩进表示