**今天的工作**

熟悉使用git，读代码，了解流程，理框架

**遇到的问题**

关于管道，重定位

**解决的问题**

1. shell是命令解释器
2. 文件描述符：一个索引，通过索引找到对应文件。

例如当进程需要从标准输入读取数据时就会通过0索引找到标准输入对应的内存缓冲区

* 1. 标准输入——0
  2. 标准输出——1
  3. 标准错误——2

1. 流程：
2. 监听用户输入命令并读取
3. 对用户命令进行解析
   1. 内建命令（我们自己编写）
   2. 外部命令（系统bin目录里，在子进程代码中调用exec族函数）——经由shell分析通过linux内核创建新的子进程，在新的子进程中执行。（在新的进程中所执行的代码是不属于 shell 的，所以在 shell 加载时并不随之一起被加载到内存中，而是在外部命令执行时才将其调入内存中。）
4. **待解决**
5. 什么是管道（引用https://www.tomorrow.wiki/archives/956）

管道：进程与进程之间通信的通道，例如可以利用管道连接两条命令ls | grep main.c

* + 1. 创建管道 pipe() 需要包含头文件unistd.h

int fd[2];

pipe (fd); //fd[0]读端文件描述符，fd[1]写端文件描述符

* + 1. 创建子进程（此例中父进程为ls，子进程为grep），子进程会继承管道，于是父子进程操作同一个管道
    2. 如ls | grep main.c就是将ls的标准输出写到管道写端，grep的标准输入从管道读端读取

1. 标准输入从管道读端读取

利用dup2(fd[0],0)将fd[0]指向的文件表项赋给文件描述符0即进程默认的标准输入。当进程需要从标准输入读取数据时，进程就会通过文件描述符 0 来找到管道读端所对应内存缓冲区。

1. close()关闭管道原有的文件描述符而不是关闭管道
2. 什么是文件重定向

**关键代码**

**其他想法**