**本HTTP服务器使用了epoll模式**

1.请求方式分为GET和POST两种：

GET：

携带参数：执行cgi脚本；

不携带参数：返回静态页面；

POST：

执行cgi脚本；

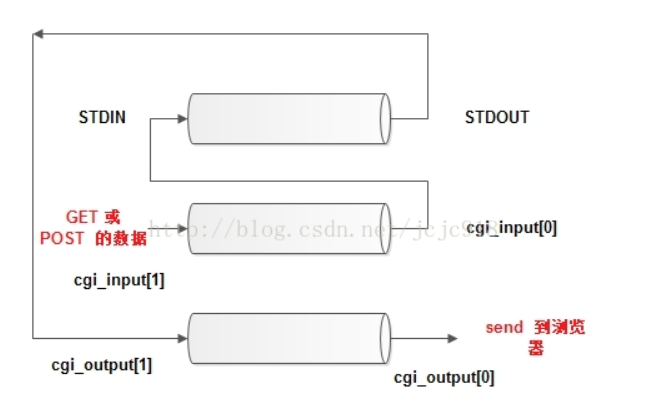
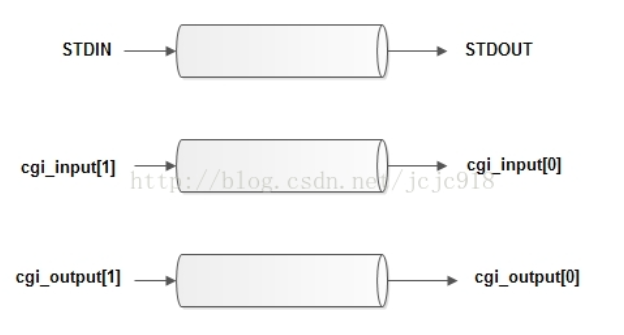
2.请求参数存放在query\_string中，长度存放在content-length中；

3.执行cgi脚本：

使用pipe创建管道用于父子进程之间通信，

4.我们所请求的数据并不在环境变量QUERY\_STRING中，用户提交的数据都存放在标准输入中，同时数据的长度是存放在环境变量CONTENT\_LENGTH中的；

5.要获取POST方式提交的数据，需要首先从环境变量CONTENT\_LENGTH中获取数据长度，然后再从标准输入中去读取该长度个字节；



6. 使用dup2(cgi\_input[0], 0)；dup2(cgi\_output[1], 1)这两个函数：

原本是从STDIN来输入，现在从cgi\_input[0]来输入数据；

原本是从STDOUT来输出数据的，现在从cgi\_output[1]来输出数据；

7.子进程需要用新的程序来代替，所以子进程的操作是用来存储环境变量的；

8.sendfile函数：直接在两个文件描述之间传递数据，完全在内核重男操作，从而避免了内核缓冲区和用户缓冲区之间的数据拷贝，效率高，被称为零拷贝；

#include<sys/sendfile.h>

ssize\_t senfile(int out\_fd,int in\_fd,off\_t\* offset,size\_t count);

in\_fd参数是待读出内容的文件描述符，out\_fd参数是待写入内容的文件描述符。offset参数指定从读入文件流的哪个位置开始读，如果为空，则使用读入文件流默认的起始位置。count参数指定文件描述符in\_fd和out\_fd之间传输的字节数；

8.为什么需要建立连个管道：

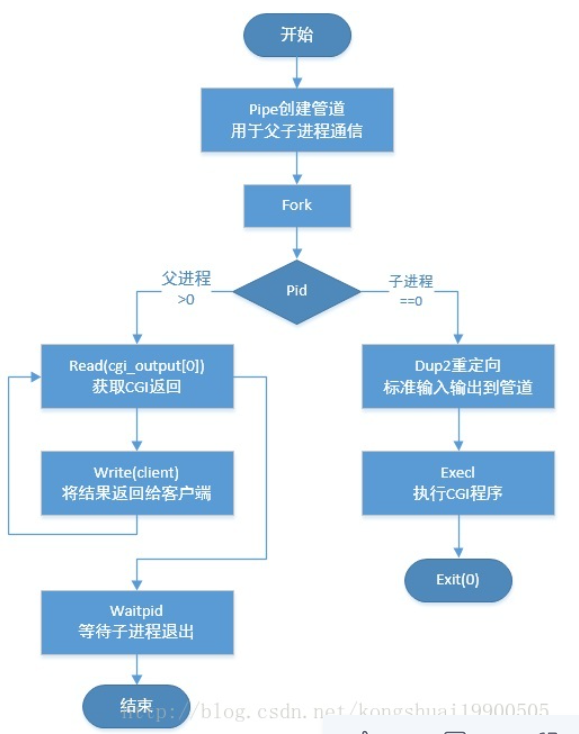
因为管道是单工通信，只能一个方向进行数据的读写，要么是读，要么是写，所以才需要两个管道相互连接，如上图所示；也就是一个管道的创建只能有一个进程负责读，一个进程负责写，双向通信需要建立两管道；

9.子进程中：

把 STDOUT 重定向到 cgi\_output的写入端，把 STDIN 重定向到 cgi\_input 的读取端，关闭 cgi\_input 的写入端 和 cgi\_output 的读取端，设置 request\_method 的环境变量，GET 的话设置 query\_string 的环境变量，POST 的话设置 content\_length 的环境变量，这些环境变量都是为了给 cgi 脚本调用，接着用 execl 运行 cgi 程序。

10.父进程中：

关闭 cgi\_input 的读取端 和 cgi\_output 的写入端，如果 POST 的话，把 POST 数据写入 cgi\_input，已被重定向到 STDIN，读取 cgi\_output 的管道输出到客户端，该管道输入是 STDOUT。接着关闭所有管道，等待子进程结束;



11.如果GET方式携带参数，那么请求url最后一定不是‘/’，因为携带参数的话，会带上‘？’，而不是‘/’，所以最后末尾是‘/’的话，那就一定是不带参数的GET方法；

12.子进程使用来处理请求的，而父进程是将结果发送给客户端的；

13.如果是在父进程中且是post方法，则需要接收客户端发送的数据，如果接收到子进程发送的数据，则将数据发送给客户端，等待子进程结束；

14.请求url中携带‘？’，那么一定是携带参数的，需要解析出来；

15.用stat函数取得文件状态，如果失败的话，就说明文件不存在或者没有权限访问，向客户机回复not\_found。正常的话则判断是否对文件具有可执行权限，可执行则让cgi = 1，如果cgi为０，则向客户端回送文件，为１则执行文件。

16.如果客户端的请求方法是GET，那么需要将客户端发送的剩余的请求头部信息全部读出来并丢弃，如果还POST方法，那就需要获取Content-length的值，也就是客户端发送过来的数据的长度，但是这个数据并不存储在url中，而是存储在请求体中；

17.父进程用于和客户端之间的数据传输(读写)，子进程通过cgi\_input接收父进程接收过来的数据，进行数据处理后，将要发送的信息通过cgi\_output传给父进程，父进程再发送给客户端，也就是父进程进行数据读写，子进程负责处理；

18.页面访问不了，需要修改权限，600或者755