咕泡教育
—gupaoedu—

分布式架构下的通信问题

GUPAO EDU

讲师：Mic

我们的愿景
推动每一次人才升级

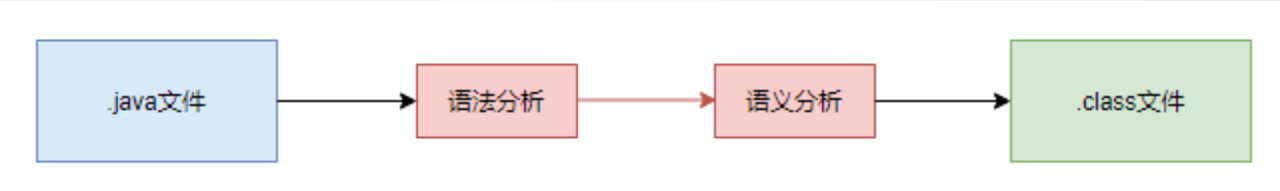
我们的使命
让每个人的职业生涯不留遗憾

本节课程内容安排

1. 理解通信的本质
2. TCP/IP通信原理分析
3. 简述TCP可靠性通信
4. 理解阻塞通信的本质
5. NIO多路复用实战


客户端
www.gupaoedu.cn
index.html
服务端

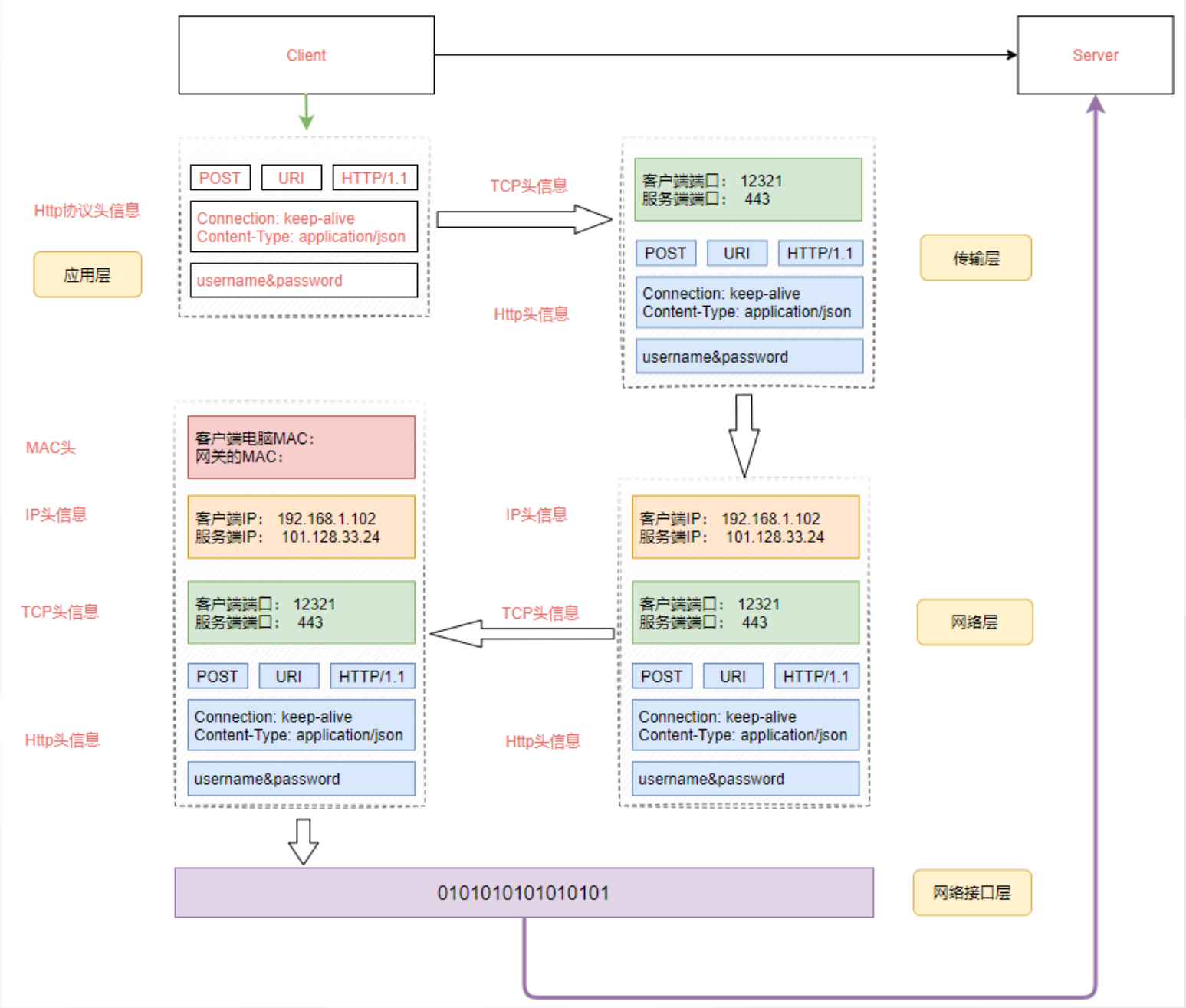
协议的组成

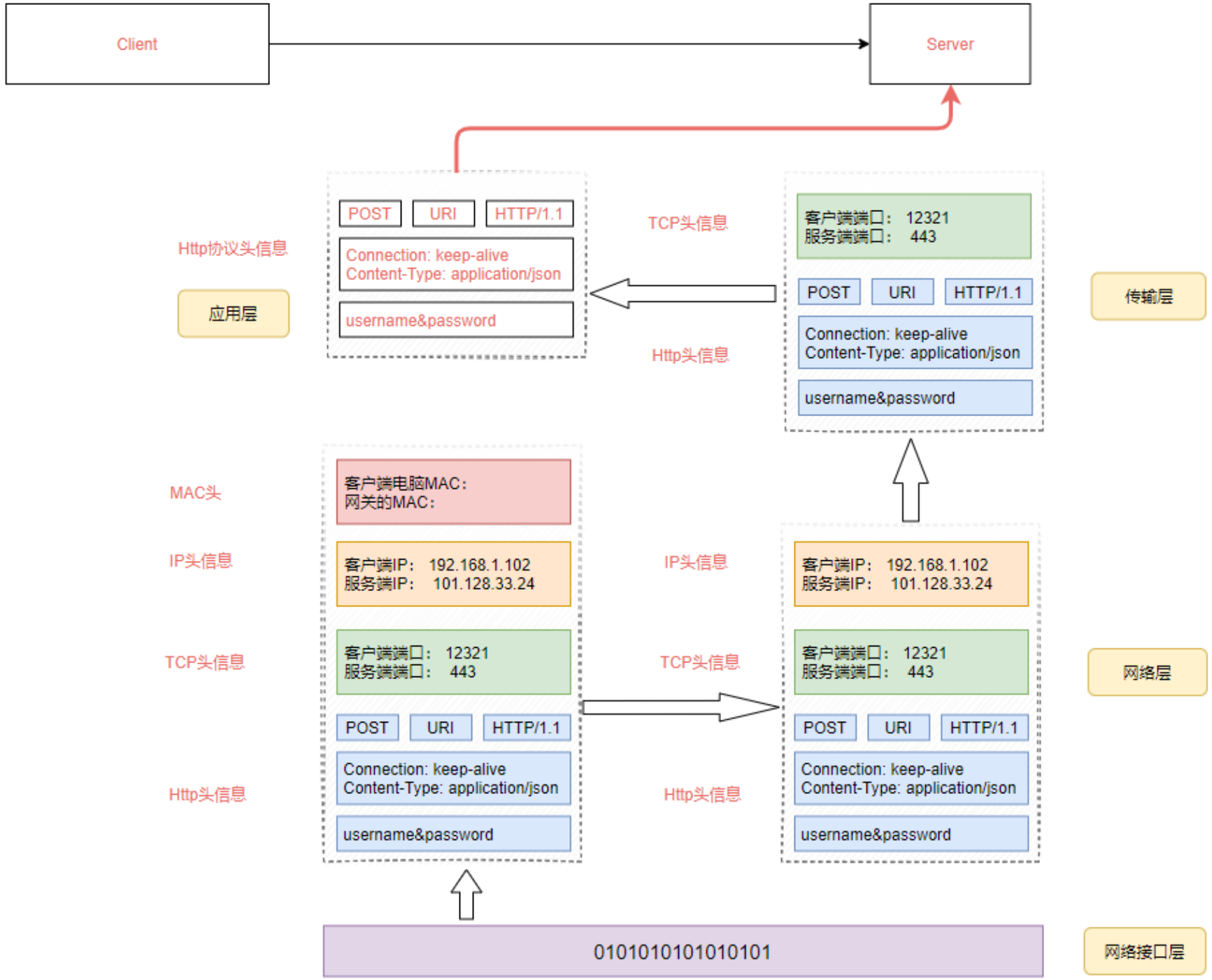


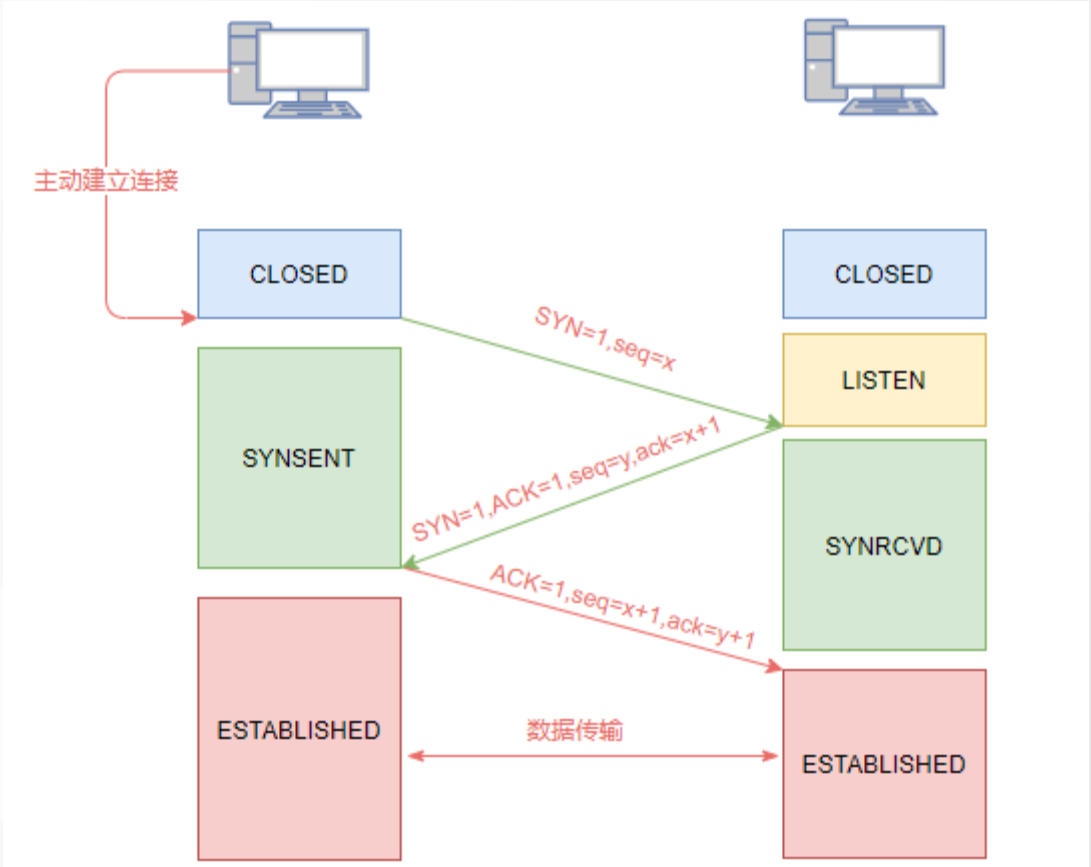
Http协议

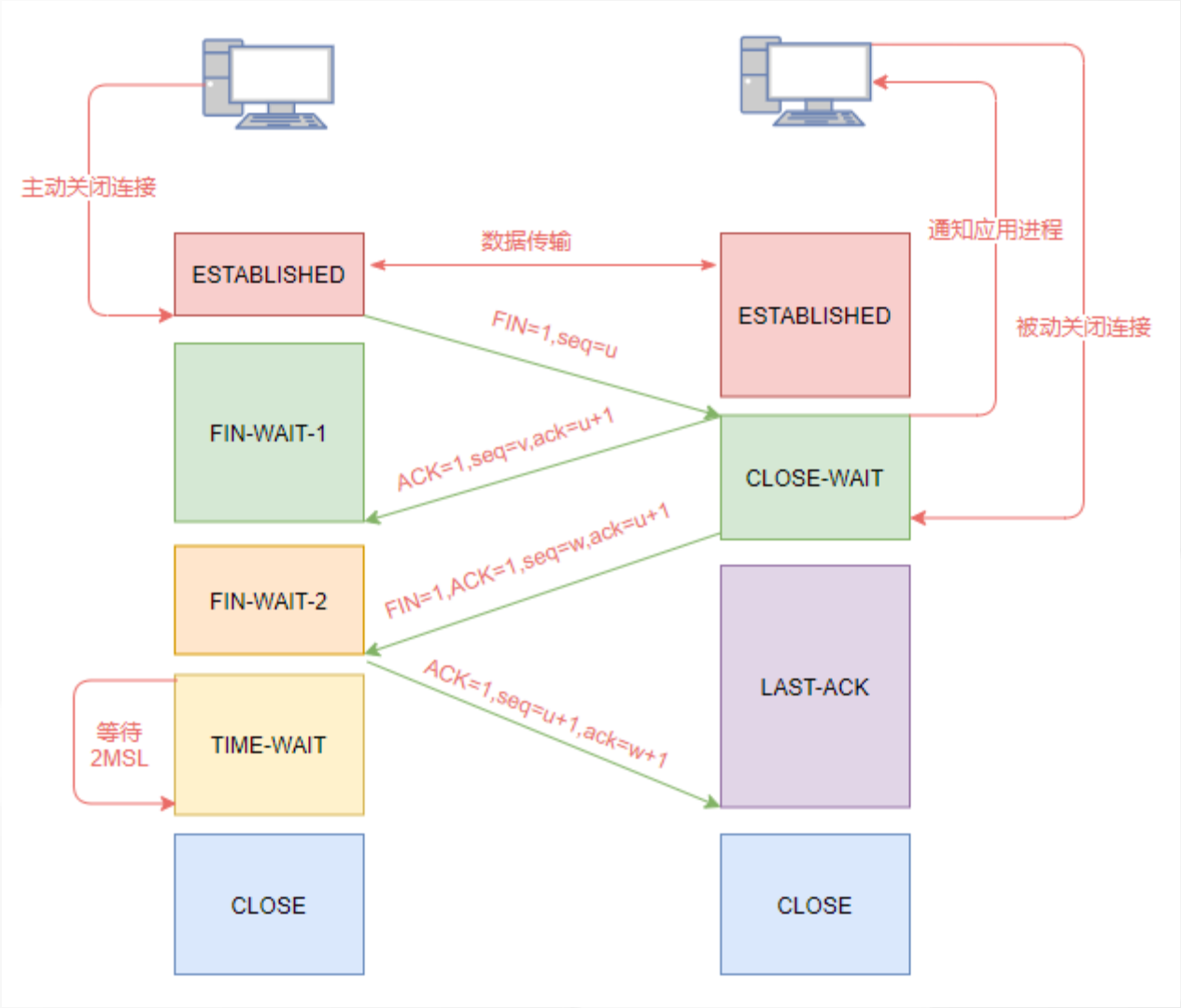
```
HTTP/1.1 200 OK                    # 下面是HTTP响应
Accept-Ranges: bytes
Cache-Control: max-age=86400
Connection: Keep-Alive
Content-Length: 81
Content-Type: text/html
Date: Fri, 02 Aug 2019 10:07:15 GMT
ETag: "51-47cf7e6ee8400"
Expires: Sat, 03 Aug 2019 10:07:15 GMT
Last-Modified: Tue, 12 Jan 2010 13:48:00 GMT
Server: Apache

<html>
<meta http-equiv="refresh" content="0;url=http://www.baidu.com/">
</html>
```

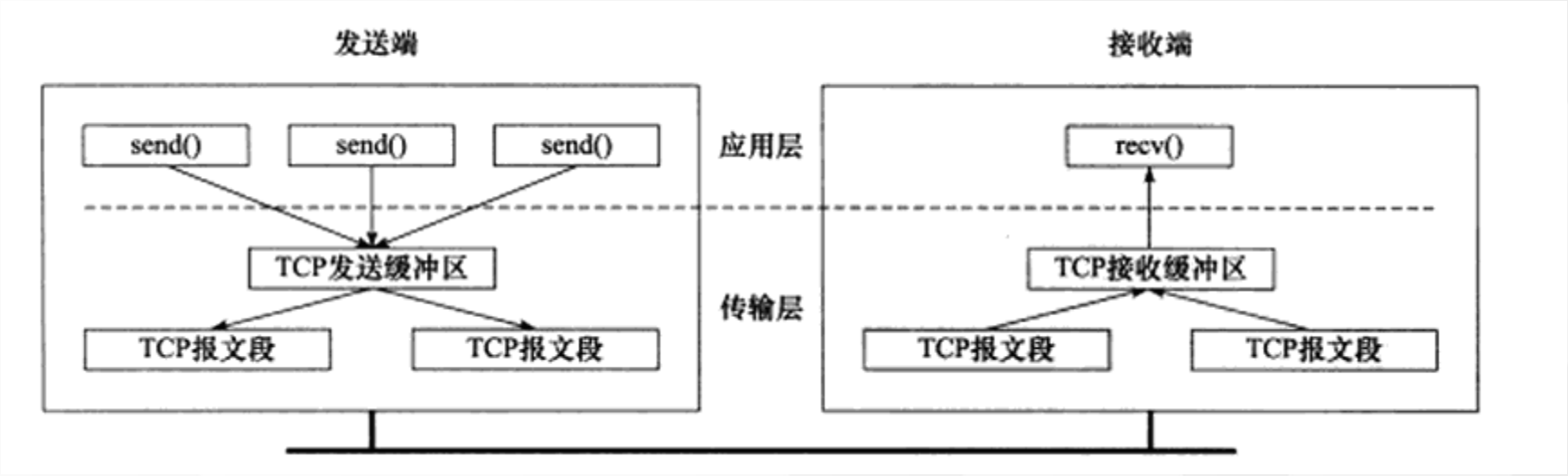

TCP/IP四层概念模型
OSI七层网络模型
应用层
传输层
网络层
网络接口层
应用层
表示层
会话层
传输层
网络层
数据链路层
物理层
为应用程序提供服务
数据格式化、数据加密
建立、管理和维护会话
建立、管理和维护端到端的链接
IP选址和路由
提供介质访问和链路管理
物理层


Client
Server
POST
URI
HTTP/1.1
Connection: keep-alive
Content-Type: application/json
username&password
Http协议头信息
应用层
TCP头信息
客户端端口： 12321
服务端端口： 443
Http头信息
传输层
MAC头
客户端电脑MAC：
网关的MAC：
IP头信息
客户端IP： 192.168.1.102
服务端IP： 101.128.33.24
网络层
0101010101010101
网络接口层

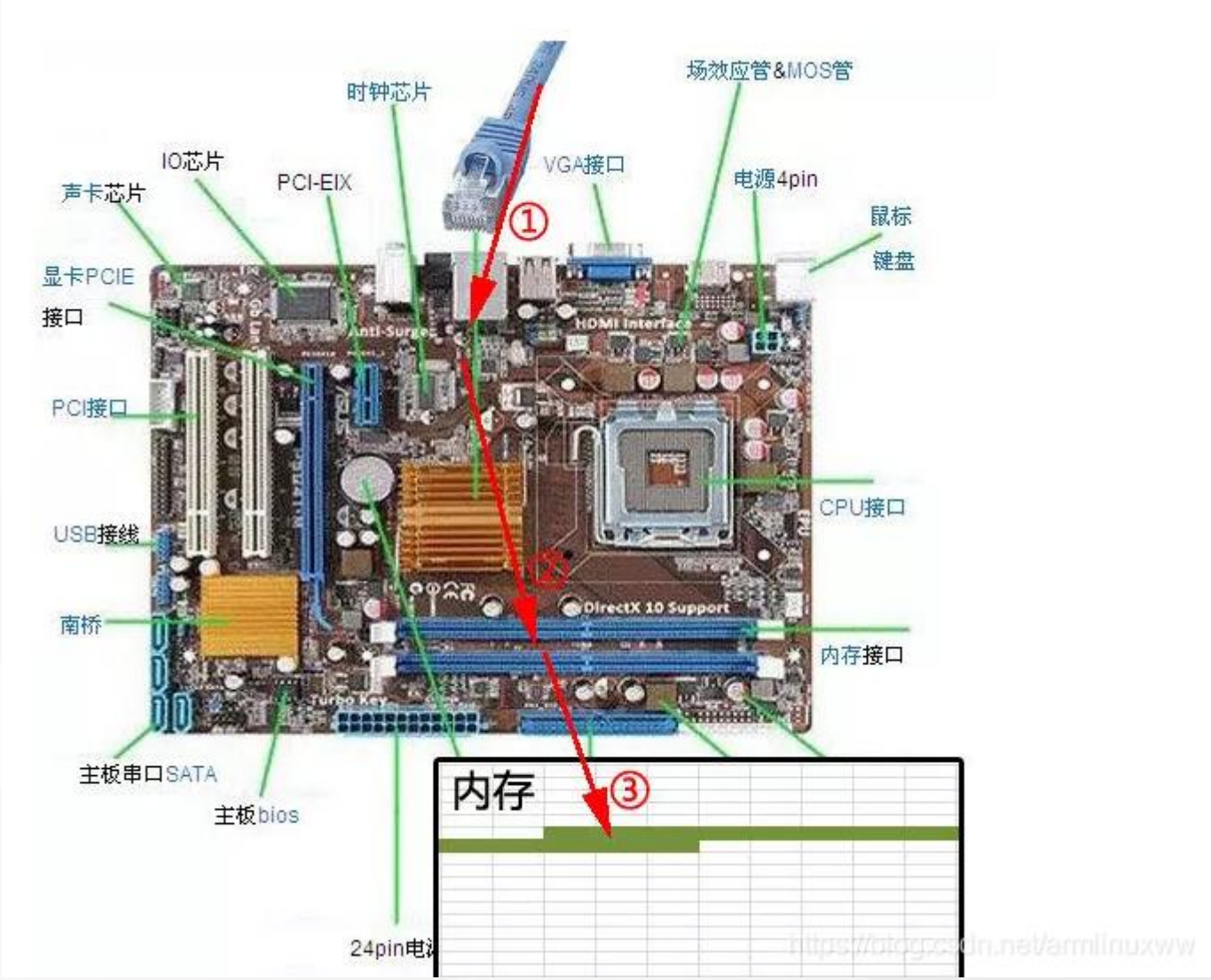

Client
Server
POST
URI
HTTP/1.1
Connection: keep-alive
Content-Type: application/json
username&password
Http协议头信息
应用层
TCP头信息
Http头信息
客户端端口： 12321
服务端端口： 443
传输层
MAC头
客户端电脑MAC:
网关的MAC:
IP头信息
客户端IP： 192.168.1.102
服务端IP： 101.128.33.24
网络层
0101010101010101
网络接口层

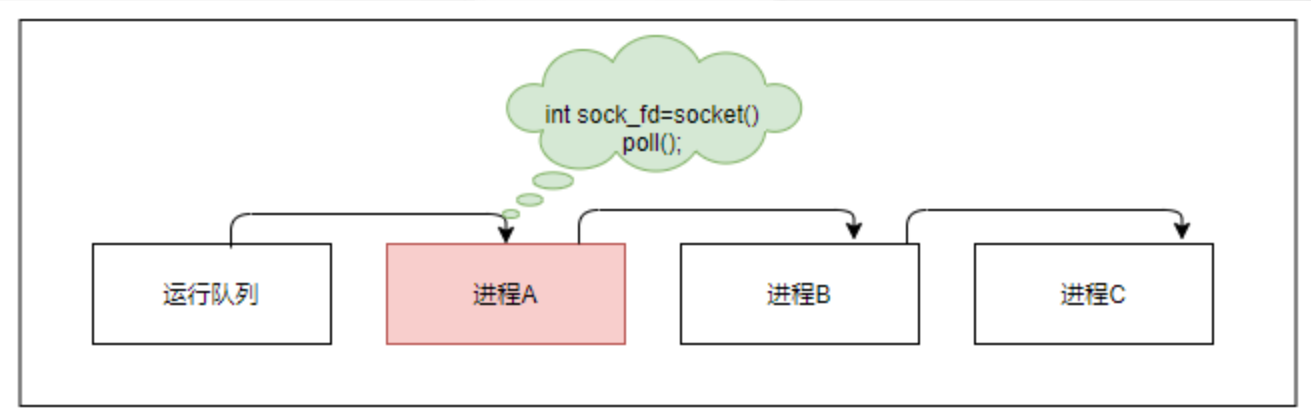

主动建立连接
CLOSED
CLOSED
SYN=1,seq=x
LISTEN
SYNSENT
SYN=1,ACK=1,seq=y,ack=x+1
SYNRCVD
ACK=1,seq=x+1,ack=y+1
ESTABLISHED
数据传输
ESTABLISHED

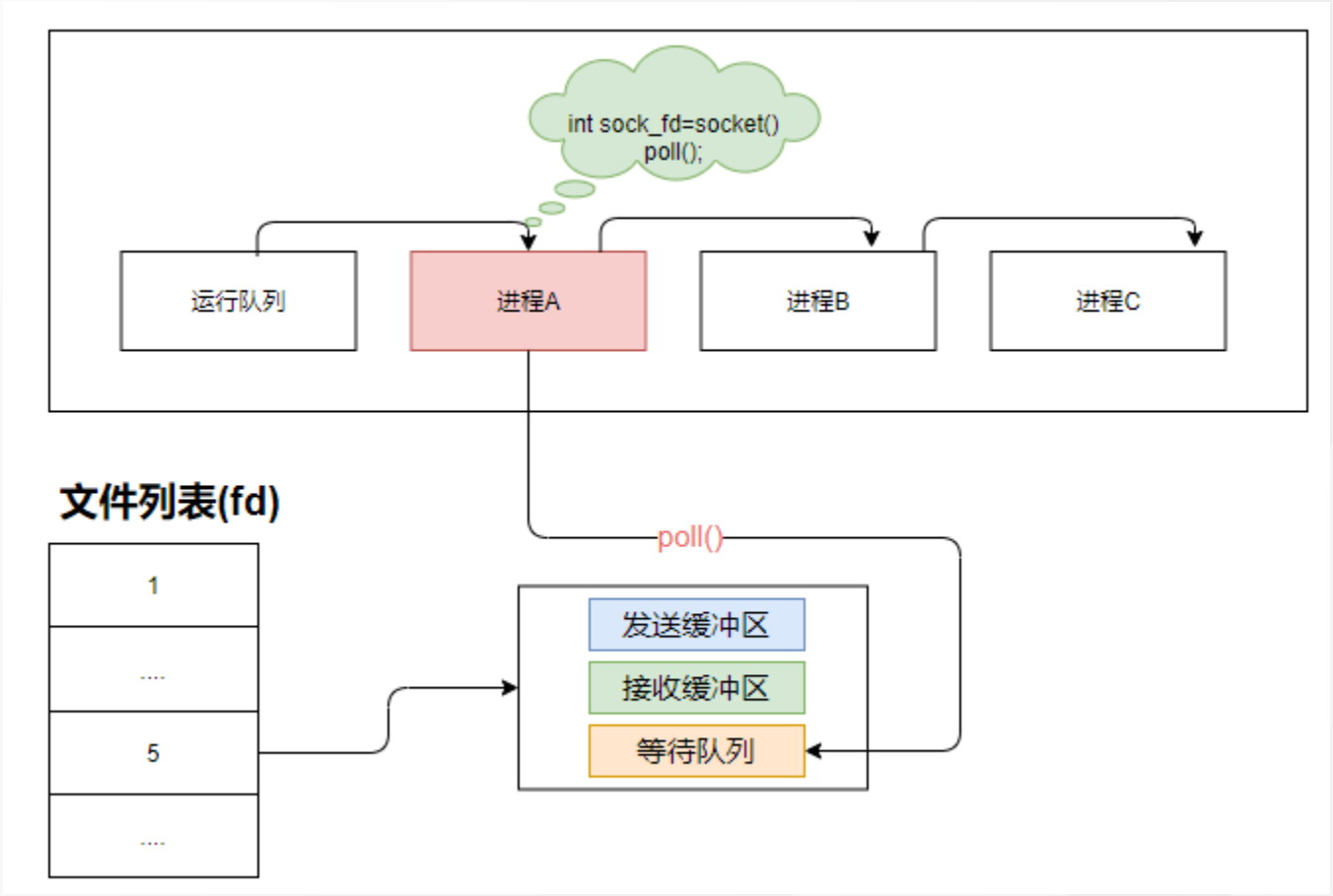

主动关闭连接
ESTABLISHED
数据传输
ESTABLISHED
通知应用进程
FIN=1,seq=u
被动关闭连接
FIN-WAIT-1
ACK=1,seq=v,ack=u+1
CLOSE-WAIT
FIN=1,ACK=1,seq=w,ack=u+1
FIN-WAIT-2
LAST-ACK
ACK=1,seq=u+1,ack=w+1
等待
2MSL
TIME-WAIT
CLOSE
CLOSE

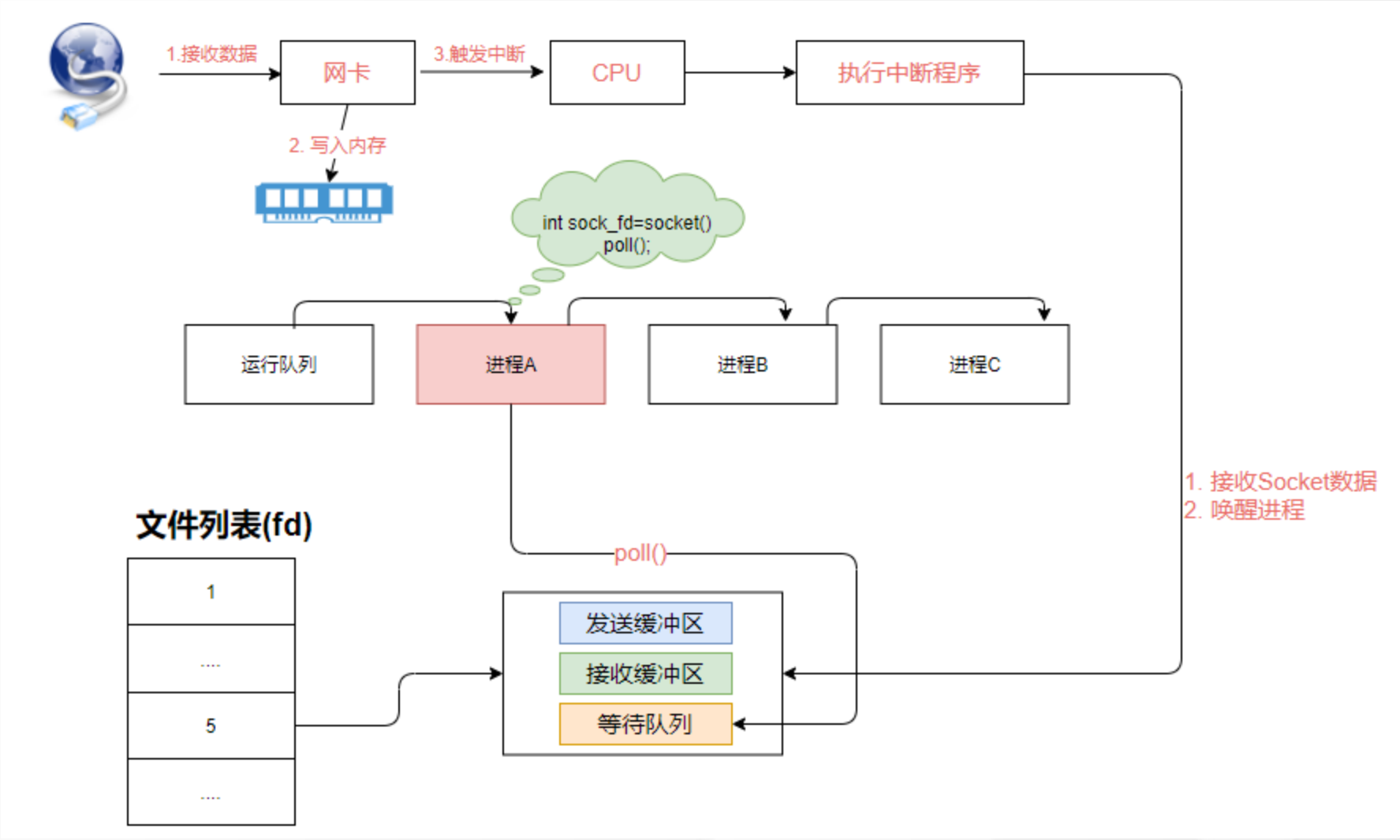

发送端
接收端
send()
send()
send()
应用层
recv()
TCP发送缓冲区
传输层
TCP接收缓冲区
TCP报文段
TCP报文段
TCP报文段
TCP报文段

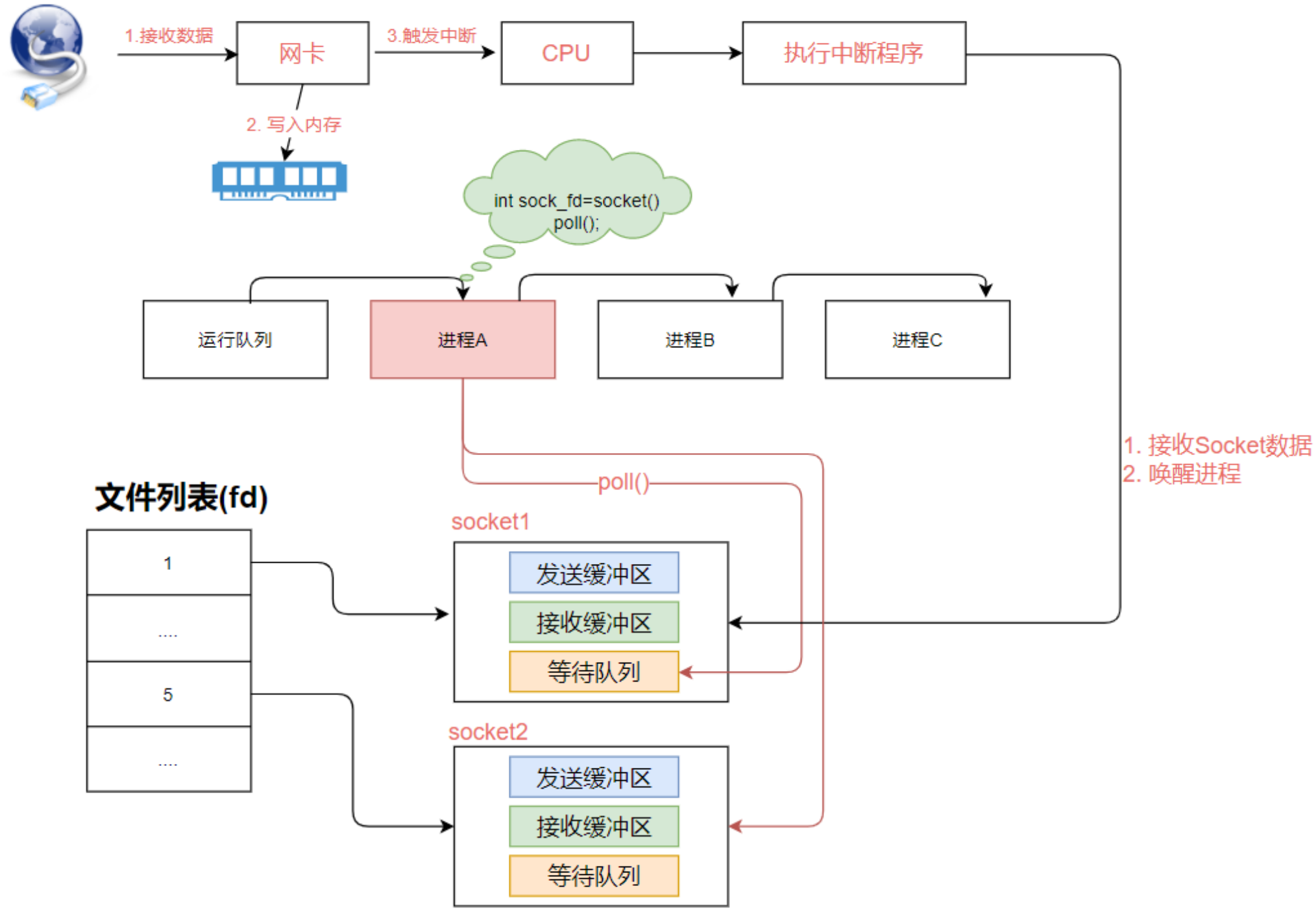

发送未确认部分
不可发送部分
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
发送已确认部分
未发送可发送部分

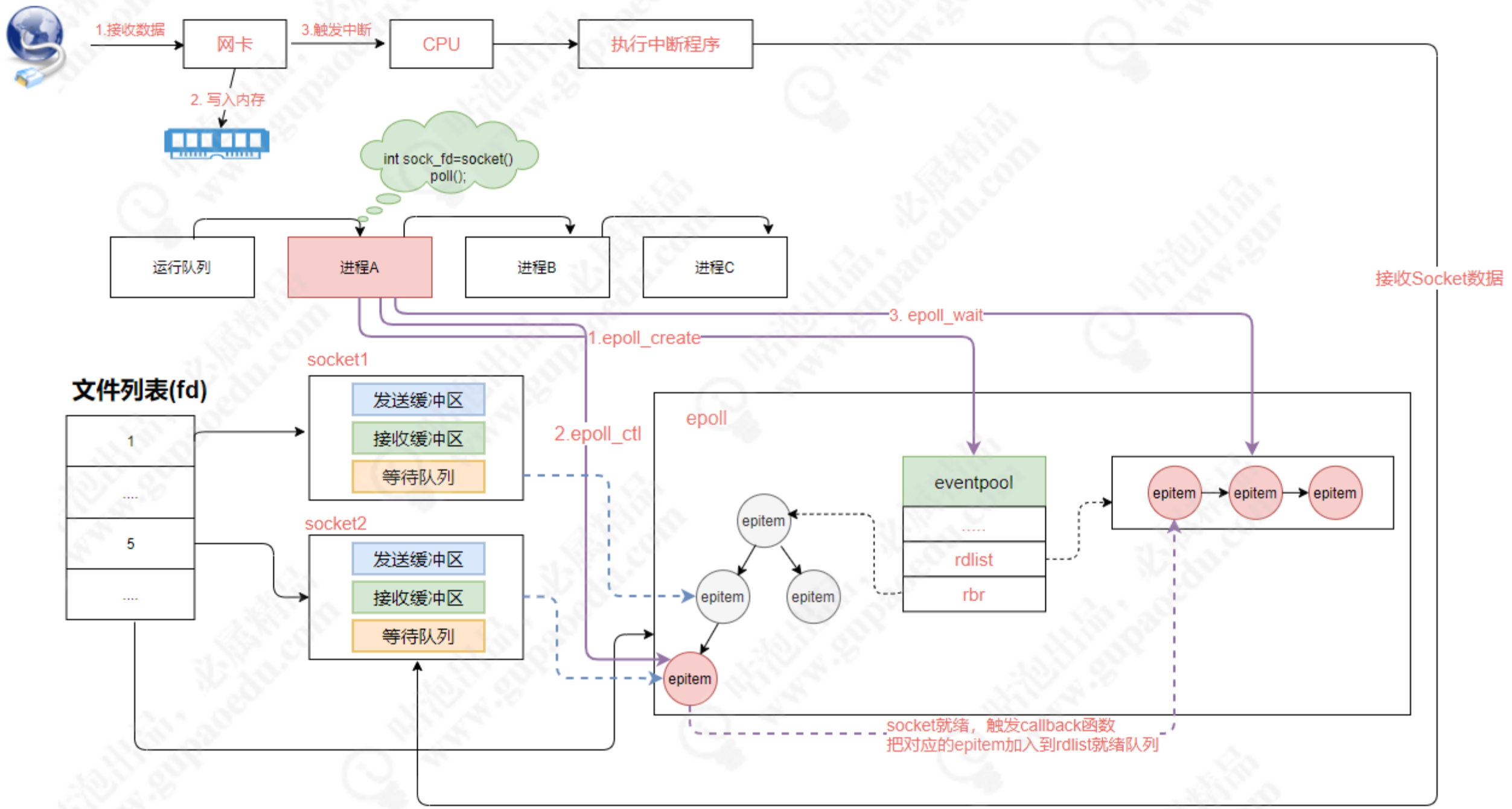

时钟芯片
场效应管&MOS管
IO芯片
VGA接口
声卡芯片
PCI-EIX
电源4pin
鼠标
键盘
显卡PCIE
接口
①
HDMI Interface
Anti-Surge
PCI接口
CPU接口
USB接线
②
DirectX 10 Support
南桥
内存接口
Turbo Key
主板串口SATA
主板bios
内存
③
24pin电


int sock_fd=socket()
poll();
运行队列
进程A
进程B
进程C


int sock_fd=socket()
poll();
运行队列
进程A
进程B
进程C
文件列表(fd)
poll()
1
....
5
....
发送缓冲区
接收缓冲区
等待队列


1.接收数据
网卡
3.触发中断
CPU
执行中断程序
2. 写入内存
int sock_fd=socket()
poll();
运行队列
进程A
进程B
进程C
1. 接收Socket数据
2. 唤醒进程
文件列表(fd)
poll()
1
....
5
....
发送缓冲区
接收缓冲区
等待队列


1.接收数据
网卡
3.触发中断
CPU
执行中断程序
2. 写入内存
int sock_fd=socket()
poll();
运行队列
进程A
进程B
进程C
1. 接收Socket数据
2. 唤醒进程
poll()
文件列表(fd)
1
....
5
....
socket1
发送缓冲区
接收缓冲区
等待队列
socket2
发送缓冲区
接收缓冲区
等待队列


1.接收数据
网卡
3.触发中断
CPU
执行中断程序
2. 写入内存
int sock_fd=socket()
poll();
运行队列
进程A
进程B
进程C
接收Socket数据
3. epoll_wait
1.epoll_create
socket1
文件列表(fd)
1
....
5
....
发送缓冲区
接收缓冲区
等待队列
2.epoll_ctl
epoll
eventpool
.....
rdlist
rbr
epitem
epitem
epitem
epitem
epitem
epitem
socket2
发送缓冲区
接收缓冲区
等待队列
socket就绪，触发callback函数
把对应的epitem加入到rdlist就绪队列

咕泡教育
—gupaoedu—

谢谢观赏

GUPAO EDU

Mic

我们的愿景

推动每一次人才升级

我们的使命

让每个人的职业生涯不留遗憾