epoll:效率高

触发方式多、

本地套接字AF\_UNIX本地两个进程捡的消息通信

多任务编程：并行：指的是几个内核一起工作 并发：是指多个进程在一个内核里不停的切换运行

多进程编程

进程进程状态：就绪态，运行态 等待态

ps –aux ps –ajx pstree top nice

os.fork()

os.getpid() os.getppid()

os.\_exit() sys.exit()

j僵尸进程是指子进程先退出，父进程没有处理子进程的状态。

如何避免僵尸进程的产生：

* 处理子进程退出状态

pid,status = os.wait()

功能 ：在父进程中阻塞等待处理子进程退出

返回值：pid退出的子进程的PID号

status 获取子进程退出状态

pid,status = os.wait(pid,option)

功能 ：在父进程中阻塞等待处理子进程退出

参数：pid -1表示等待任意子进程退出

>0表示等待对应pid号的子进程退出

option 0表示阻塞等待

WNOHANG表示非阻塞

返回值：pid退出的子进程的PID号

status 获取子进程退出状态

waitpid(-1.0) =🡺 wait()

* 让父进程先退出

1. 父进程创建子进程等待子进程退出
2. 子进程创建二级子进程后立即退出
3. 二级子进程称为孤儿，和原来的父进程各自执行事件。

作业：

写一个聊天室：

1. 进入聊天室需要输入姓名：姓名不能重复
2. 有人进入 聊天室会向其他人发送通知，xxx进入了聊天室
3. 一个人发送消息，其他人会收到通知

xxx说：xxxxxxxxxxx

1. 某人退出了聊天室，其他人也会收到通知

xxx退出了聊天室

5．管理员喊话。服务端发送消息所有客户端都收到

管理员说：xxxxxxxxxxxx

功能模型：转发

需要的技术：套接字通信，udp套接字

用户存储：字典

消息的收发的随意性；多进程

代码设计：包含封装（先函数，后面为类），接口测试？？？（实现一步测试一步）

代码编写流程

搭建网络连接—》创建多进程---》每一个进程功能编写----》项目功能模块实现

进去聊天室：

客户端：输入姓名 将信息发给服务端（L name）

等待服务端回复 根据回复判断是否登录成功

服务端： 接收请求信息，判断请求类型：查看名字是否存在，不存在插入到数据结构，回复可以登录，存在返回说名字重复。

聊天

客户端：创建父子进程，发送聊天请求/接收聊天信息

服务端：接收请求信息，将消息转发给其他客户端

multiprocessing模块创建进程

1. 需要将要执行的事情封装为函数
2. 使用multiprocessing模块中process类创建进程对象
3. 通过对象属性设置和process的初始化函数对进程进行设置，绑定要执行的函数
4. 启动进程，会自动执行绑定的函数
5. 完成进程的回收。

process（）

功能：创建进程对象

参数：name进程名称： process-1

target绑定函数

args元组 给target函数按照位置传参

kwargs字典 给target函数按照键值对传参

p.start()

功能： 启动进程

* target函数会自动执行，此时进程真正被创建

p.join([timeout])

功能：阻塞等待回收子进程

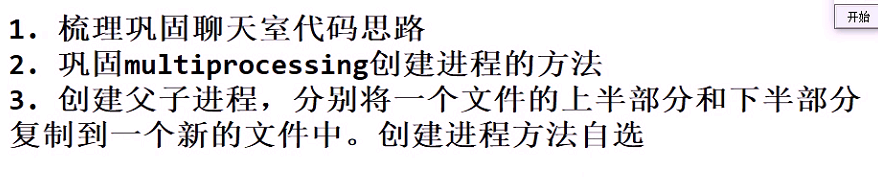
参数： 超时时间

\*使用multiprocessing创建子进程，同样子进程赋值父进程的全部代码段，父字进程各自执行互不互不影响，父子进程有个字的运行空间

\*如果不适用join回收子进程则子进程退出后会成为僵尸进程

\*使用multiprocessing创建子进程往往父进程只是用来创建进程回收进程

如果子进程使用父进程的对象，那么相互之间有偏移量的影响



打开文件，读取数据，拆分数据，father写上，son下

拆分数据，字节串，len()总数，当为偶数l%2==0切片【0:int(len(a)//2)+1】奇数【int(len(a)//2)+1：】

读文件：fr=open(‘wenji

file=input()

1.1

1.

def read(file):

fr = open(file,’rb’)

s= f.read()

fr.close()

return s

def write(s[]):

fw = open(file.’wb’)

fw.write(s[])