

---

1 进程间通信机制有哪些？ 管道、信号、共享内存、消息队列、信号量、socket

效率最高的是？ 共享内存(因为共享内存采用的是映射机制，两个进程可直接访问)

共享内存有什么缺点？ 访问共享内存时，容易产生冲突（解决办法：用互斥锁）

有先进先出机制的是？ 管道、消息队列

假设要实现类似于 QQ 的软件：用哪种机制 缓存消息？ 消息队列

2 线程间怎么进行数据传输(线程间通信机制有哪些)

(1) 用全局变量 （因为多线程全局区共享）

(2) 线程也可以采用进程间通信的机制(管道、共享内存、消息队列、信号量、socket)

3 线程和进程的区别是什么？ 如何创建线程如何创建进程？ 如果将一个值传递给线程可以用什么方法？

(1) 进程是一个独立运行的程序，但是线程不是，线程要依附于进程，进程结束，线程也结束

(2) 进程创建时要新建 所有内存区，单线程只新建栈区，所以从效率角度来说，线程效率更高

创建进程 fork

创建线程 pthread\_create

给线程传递值：用 pthread\_create 的第四个参数，类型是 void \*

4 阻塞和非阻塞？ 说一下有哪些场合有阻塞机制

阻塞： 程序执行到某处，停下来(等待某个条件)

非阻塞： 条件不满足，程序也不停

scanf

sleep

wait （父进程等待子进程结束，一旦子进程结束，父进程解除阻塞）

accept （服务器阻塞等待客户端的连接）

recv （等待接收数据）

sem\_wait （请求信号量）

mutex\_lock （加锁失败，会阻塞）

read （读管道，没有数据会阻塞）

5 线程间的同步方式(多线程中信号量和互斥锁用来解决什么问题？)

可以用信号量实现同步，

可以用互斥锁或信号量 实现互斥

---

6 I/O 文件读写 -> 具体指出一下(描述就可以)

文件 I/O , 文件读写

用过两类函数

- 1) 标准 c 文件 I/O (fopen fread fwrite fgets fseek fclose)
- 2) linux 文件 I/O (open read write lseek close)

7 linux 文件 I/O 和 标准 c 文件 I/O 的区别

- 1) 标准 c 文件 I/O 带缓冲区机制, linux 文件 I/O 不带缓冲区
- 2) 使用方法不同 (标准 C 文件 I/O fopen .... linux 文件 I/O open ....)

华清远见沈阳分中心