

# Linux 期末复习

## 简答以及代码题类型可能知识点

### 1. 常用的 Cat 命令有哪些？

(1) 查看一个文本文件的内容

```
cat filename
```

(2) 创建文件：将标准输入重定向到文件 filename，覆盖文件

```
cat > filename
```

(3) 连接文件：将 f1 和 f2 内容合并到 f3

```
cat file1 file2 > file3
```

(4) 显示多个文件内容

```
cat file1 file2
```

(5) 追加内容到文件：将标准输入追加到 file 末尾

```
cat >> filename
```

## 2. 文件锁(互斥锁)的功能概念，文件锁的上锁规则

(1) 文件锁是应用在多个线程竞争访问同一个文件的时候，方便多个进程能够顺畅访问到最终的文件所采用的一个同步方式。

(2) 读锁可以继续上读锁；读锁不能被上写锁；写锁不能上任何类型的锁

## 3. 守护进程的概念，创建守护进程的步骤。

(1) 守护进程是系统后台长期运行的一个服务进程

(2) 创建守护进程的步骤：

- 创建子进程，父进程退出
- 在子进程中创建新会话
- 改变当前目录为根目录
- 重设文件权限掩码
- 关闭文件描述符

## 4. 管道(pipe)的理解

管道分为 3 种：无名管道、标准流管道、有名管道

无名管道(pipe)：

- 只能用于具有亲缘关系的进程之间的通信(父子进程和兄弟进程之间)
- 半双工的通信模式，具有固定的读端和写端
- 管道不是普通的文件，不属于其他任何文件系统，只存在于内核的内存空间中

标准流管道(含有函数 popen)：

- 固定读写端的管道，写端固定于键盘，读端固定于显示器-----特征
- popen 函数是标准流管道中集成了一切动作的函数，其效果如此

有名管道(mkfifo):

- 可以使两个不相关的两个进程实现彼此通信
- 其严格遵循先进先出(FIFO)的规则
- 区别于无名管道只存在于内核中,有名管道有个实体的文件名存在于我们的磁盘上

注意区分和分辨 3 种管道的特性

## 5. 信号量

- 信号量对应一个临界资源,二维信号量只有 2 个值,非 0 即 1
- 等于 0 时只能执行 v 操作, v 是+1 操作
- 等于 1 时只能进行 p 操作, p 是-1 操作

信号量 fork.c 猜题:

1. 描述这是什么功能的代码(特征: fork 函数, 父子进程)
2. 信号量相关代码, 删除和保存的运行结果不同

删除: 子进程在运行中---父进程先结束---子进程后结束

保留: 子进程在运行, 父进程等待结束---子进程结束---父进程结束

## 6. 共享内存

书上 p260 页, 图 8.8 注意理解

- a. 共享内存被称为是最为高效的进程间通信的手段, 因为数据不需要进行复制, 在内核的共享内存区域, 把这个区域映射到两个进程各自的地址空间当中, 这样的话两个进程对自己空间内的数据进行读写, 从而间接实现通讯间的效果。
- b. 共享内存实验使用信号量来约束读写操作:
  - i. 生产者与消费者实验(我也不太懂, 自己慢慢看)

## 7. 多线程

(1) 3 个函数：

- `pthread_create()`
- `pthread_join()`
- `pthread_exit()`

书上 285 页，具体代码自己看下

(2) 互斥锁线程控制和信号量线程控制（主要内容在信号量，易考辨析和信号量线程控制的原理）

互斥锁辨析：

- 加锁：线程在进入临界区前，会尝试获得互斥锁。若锁已被其他线程占用，则当前线程将被阻塞，直至锁被释放。
- 执行：一旦获取到互斥锁，线程进入临界区，执行需要互斥访问的操作。
- 解锁：线程离开临界区时释放互斥锁，允许其他线程进入。

信号量辨析：

- 详解：信号量机制是一种用于进程同步和互斥的方法，在多个进程同时访问临界资源时，可以使用信号量来控制进程的访问。信号量是一个计数器，用于记录可用资源的数量。
- 当进程需要访问临界资源时，它会尝试获取信号量。若信号量值大于 1，则进程可以访问资源，并将信号量-1；若信号量值为 0，则进程需要等待，直到其他进程释放资源并增加信号量的值。

## 8. 网络编程

a. socket 基础编程、高级编程

- i. 区分：一个 server 对一个 client；一个 server 对多个 client

- b. 一个服务器监听多个客户端数据
  - i. `select()` 进行监控, p313 代码

## 学习知识点

### `waitpid.c` 相关知识点

- a. 该程序作用是阻塞父进程, 等待子进程结束; 子进程不结束, 父进程不能先结束
- b. 进入 GDB 调试, 默认进入父进程调试, 子进程不受影响
- c. 调试父进程的设置: `set follow-fork-mode parent;`
  - i. 子进程顺利执行: `set detach-on-fork on;`
- d. 调试子进程的设置: `set follow-fork-mode child;`
  - i. 父进程不执行: `set detach-on-fork off;`
- e. 如果调试子进程, 父进程先阻塞, 然后再进入新一轮调试父进程
- f. `waitpid(pid, NULL, 0)`, 此函数作用是阻塞父进程

## 信号量机制通过共享内存实现的生产者(pr)与消费者(cu)问题

- a. 首先必须创建共享内存区域, 然后分别映射到 pr 进程和 cu 进程
- b. 信号量初值必须是 0
- c. 初值为 0 的信号量不能执行 v 操作, 否则会失去约束条件
- d. cu 进程的第一条语句必须添加 p 操作
- e. pr 进程的第一条语句不需要添加 p 或 v 操作

- f. pr 进程的最后一条语句必须添加 v 操作，才能让 cu 进程有机会执行
- g. 信号量机制实现双线程的通信，可以使 cu 进程先运行，pr 进程后运行，仍能得到预期效果

## 互斥锁和信号量约束线程执行次序

- a. 互斥锁和信号量都可以约束线程按照指定次序运行
- b. 3 线程下，需要 1 个互斥锁，或 3 个信号量，即可实现多线程的指定次序运行
- c. 互斥锁只能有 1 个，而信号量可以有多个
- d. 先后创建的线程进入临界区，其运行次序来自于信号量的约束

## socket 编程(ABCDE)

20. (多选题, 10.0 分) 下面关于socket编程的说法中，正确的是:

- A. socket()函数用于建立一个socket连接，对sockaddr或sockaddr\_in结构体进行初始化;
- B. bind()函数用于将本地IP地址和端口号实现绑定，但在UDP连接中没必要。
- C. accept()函数用于服务器接收客户端发起的socket连接请求。
- D. connect()函数用于客户端向服务器主动发起socket连接。
- E. send()函数和recv()函数的效果与write()函数和read()函数对同一个文件的读写操作类似。