文件、文件夹，设置文件访问权限等基础问题

第一章第二章是使用Linux的基础，必须要会并且掌握

这个课程主要是Linux环境下的编程开发

不考核，通过后面简答题和程序阅读题来检验是否掌握前面的基本命令

A：核心内容 A： 必须掌握 A：记下来 A：知道了解 A：可忽略 A：听不清楚

# 第三章

P53

P54 C语言编译过程的图（流程，图要记下来）

P55 vi编译器（必须要掌握）三个模式：插入模式（输入字符）、命令模式（编辑字符）、末行模式（保存、模式匹配、查找指定字符串） 三个工作模式及每个工作模式的主要功能，必须要记下来

P60 gcc编译流程解析的四个步骤

预处理：是把一些宏定义、宏函数包含进去了

编译：把源文件进行语法规则检验后，编译成汇编语言

汇编：将汇编语言翻译成目标文件 .o

链接：把一些函数库链接到可执行程序里，从而形成二进制的可执行文件

这四个步骤与图3-1命令过程基本上是类似的

这四个编译过程要记下来，以及每个过程要完成的功能动作要记下来

P64 函数库是链接阶段用到的

函数库：编译系统，把用户常用的一些功能事先开发出来，然后用户想要用的时候直接调用就可以了

函数库分为静态库和动态库

静态库：不管运行还是不运行，直接在链接阶段把所有所需的文件都打到可执行程序里

动态库：在编译运行的时候再链接进去

静态库、动态库概念必须要掌握，过程不要求，但是名词概念必须区分的开

P67 gdb调试器重点考核内容

为了能够读懂代码，必须会用调试器把程序一下子执行的过程停顿下来，一行一行的按照它中间执行的状态，这样的话就能够读懂程序的代码

gdb调试 以源文件test.c为例 需要掌握哪些

最开始，编译带有调试的可执行文件的命令： gcc -g test.c -o test

进入调试：gdb test

首先敲l（list）显示程序代码的行号

b 10 给第十行设置一个断点

设置好断点后，敲r run运行，停顿到断点行

p n 将n这个变量打印出来

打印的意义：看实际执行是否和预期一样，看你是否读懂代码

继续执行有两种方法：一种是单步s（step），还有一种逐函数执行n（next），如果接下来是一个函数调用的语句，它直接把函数调用的结果执行出来，如果对函数执行细节不清楚的话，就直接把结果执行出来n和s，然后再敲s，一步一步往下执行

如果底下所有语句都明白，应该敲c（continue），一下子就执行完了

在调试过程中还有一些其他的功能，比如给变量赋新的一个值，断点可以设置到函数定义的位置 、显示断点 info b 、删除断点

需要掌握的就是上面口述的这几条，必须要进行记下来，这是重点考核内容

P75 make工程管理器（的功能、作用）：方便用户编译、管理工程文件

功能实现核心要点：makefile文件

依赖关系用makefile进行描述

P76必须掌握 makefile文件是如何构成的

makefile的构成

一般格式：

target:dependency\_files

command/\*该行必须以Tab键开头\*/

目标文件冒号来自于依赖文件通过command的编译命令编译生成的

一般格式必须记住

command前面不是空格，是Tab键

makefile变量

知道makefile 变量是什么意思

就是用一个短的字符串来代替一个长的字符串，来减少makefile文件的编码量的一种途径，就是makefile变量

默认变量，自动变量含有不同的意思，看看就行了

P79 makefile规则，包括隐含规则，模式规则（了解一下就行）

P81make工程管理器管理工程非常简单，之所以功能能够实现是因为有makefile文件存在，所以makefile文件非常重要，手写太复杂，所以借助工具

autotools工具来自动生成makefile文件（知道这个工具干什么用，它的功能是干什么，就行了）问作用、功能

# 第六章

P153 文件I/O编程

三个概念：系统调用、用户编程接口和系统命令

系统调用：操作系统给用户提供的一些接口，通过这些接口用户可以使用系统的核心的一些服务。系统调用最终的具体实现就是通过用户使用一些API的一些函数来把这些系统调用的功能实现出来。

而系统命令在用户编程接口之上：它是可执行程序

三个名词概念，以及他们之间的层次关系必须要掌握

P154底层文件的I/O操作

用到的函数有open() 、read()、write()、lseek()这些常用的文件I/O函数

P157 copy\_file.c的源文件

实现的功能，从src\_file文件中按照lseek()所指定的文件的起始位置，以及读它要读的多长文件区域，把源文件的内容取出来，然后拷贝到目标文件dest\_file。用到的函数open() 、read()、write()、lseek()

这个源程序不直接考，但后面的会考到，所以copy\_file.c这个源文件很重要，所以最好熟练掌握（指哪一行就能说出哪一行的作用）

比如while循环的这个函数

write（dest\_file,buff,real\_read\_len）;write函数用到的三个参数的含义：从buff这个字符数组当中把数据取出来，写到dest\_file文件当中，写real\_read\_len这个长度；read也一样

源程序很重要，必须要掌握

P158文件锁

文件锁要解决的问题就是多个进程同时访问同一个文件时，如何并发控制访问：用文件锁方式来实现

提供的类型：记录锁，可分为读锁和写锁

P160三段源代码：lock\_set.c、write\_lock.c、read\_lock.c

lock\_set、写锁、读锁的源程序

lock\_set相当于文件锁的一个函数库；核心功能：判断指定文件有没有锁、能不能上锁，其次就是给它上指定的读锁、写锁。所以相当于上锁的函数库，具体使用这个函数库的就是write\_lock和read\_lock（写锁和读锁）

这个程序执行的最终效果就是揭示了上锁的原则；上锁的原则就是，一个进程给一个文件上了读锁以后，另外一个进程可以再上读锁，但不可以上写锁；上了写锁以后也不能上其他锁；唯一能上两个锁的就一个条件，两个进程同时上读锁，可以。

文件锁的功能，要记住

上锁的结论，要记得

P163I/O多路复用 重点掌握

对应的源程序P166 multiplex\_select.c要考

考法，在全部代码当中抽取一段来考， 解释意思（带参数），要把参数说清楚

核心要点：select（）函数监控

文件描述符集包含三个文件描述符：标准输入、管道文件in1、管道文件in2、

select（）函数一直在监控，监控客户端cat > in1,敲个字符，只要一敲回车，select（）函数监控到shell窗口cat命令输入的键盘的字符，那么监控到以后，监控的文件描述符集就cat，inset,就只包含有管道文件in1的文件描述符，把in1输入的字符通过后面的流程在屏幕中打印出来（核心就是这个）看视频

课后实验P192用文件读写的方式来模拟生产者-消费者经典实验

具体的模拟效果:有一个有名管道叫myfifo,生产者向myfifo文件里面写字符abcdefg，消费者进程从管道文件中把输入字符取出，就是模拟生产者和消费者（通过文件读写的方式来实现）

这个源程序中，生产者和消费者用到的程序代码把文件I/O的操作函数read()、write()用了还用到了文件锁（谁先执行，谁后执行），多路复用没有用

因为有两个进程，没有先后次序，所以用了文件锁

copy\_file、fc\_tll讲了，那这个代码应该会

用文件读写来模拟仿真生产者—消费者经典问题源程序要考，截选代码，解释每行含义，所用参数代表什么意思

# 第七章 进程控制开发

重点是一些概念的定义，作用是为第八章进程间通信做准备

进程的基本概念：进程控制块、进程标识等等，这些必须要知道

进程运行的状态，进程的三个状态：就绪状态、等待状态、运行状态

就绪状态：运行条件满足，但是还没有轮到它运行

等待状态:运行条件不满足，等待条件满足，进入就绪状态

运行状态：放在CPU运行

三个状态必须要知道

P206 进程控制接口相关的函数：

fork()函数是唯一的执行一次有两个返回值的函数，返回值为0代表在子进程中，如果返回的是一个正整数，代表是在父进程中，这个必须要知道

P208 用fork（）函数创建子进程的意义是要让它完成指定的工作

如何完成指定工作：使用ecec()函数族来实现指定工作。这就是这个函数族的作用意义价值，把这个记住就可以了

P214waitpid()

P216waitpid.c源程序必须掌握

讲教材讲过一次，多进程gdb调试又讲了一遍

核心功能：阻塞父进程，等待子进程退出，子进程没有退出父进程不能退出

不同的状态，就是第三个参数，要么为0，要么WNOHANG

0代表阻塞父进程期间，父进程不能做任何事

WNOHANG：wait no hang 等待的时候不挂起，意思是父进程还可以做一些事，子进程检测返回值，sleep多长时间，再看子进程是否退出，程序必须要掌握

P217守护进程

守护进程的含义（必须要记得）：它是系统后台运行的进程，脱离控制终端的（必须知道）

创建守护进程的步骤（5步）：

1、创建子进程，父进程退出，使得子进程变成孤儿进程，被系统的init 1号进程收养

1. 在子进程中创建新的会话，一个会话就是一个操作的过程一个基线
2. 改变当前目录为新目录，使得访问权限不成（限）对象
3. 重设文件权限掩码
4. 关闭文件描述符

这几个步骤必须记得

P220 代码是前面文字描述的代码体现，看下过程就行了

P221 守护进程的出错处理

守护进程脱离了控制终端，所以不能用gdb进行调试，而且运行是在后台默默运行，想要看出错信息只能到系统的日志文件（可以把系统的运行状态，各种情况全部记录，通过查看日志文件可以判断守护进程是否出错）

查看守护进程是否出错的方式必须知道

# 第八章 进程间通信（重点掌握）

6种通信方式：

1、管道；管道又分成三种：无名管道、标准流管道、有名管道（三种管道的类型、特征、彼此之间的区别是什么，必须要掌握）

无名管道（特点）：在有亲缘关系的进程间使用的通信手段，无名管道的文件只存在于内核空间当中，并没有实体文件存在

标准流管道（特点）：固定定读写端，写端固定于键盘，读端固定于显示器，把创建管道的一系列动作标准化到一个标准函数popen()里，这样的话，大大缩减了书写管道的占有量

popen()有两个参数，第一个参数：shell命令，第二个参数：管道类型（读端r还是写端r）

改动的是sl(i)pt对应的shell命令

管道之间的区分、特征，必须要掌握

有名管道（特点）：创建了一个管道文件，对这个管道文件进行读写，对有名管道进行读写，不限制进程的亲缘关系，所以任意两个进程都可以进程通信（通过读写有名管道文件，间接实行通信，本质就是一个文件读写的工作）

P243 信号

信号是一种古老的通信方式，是软件层次上对中断机制的一种模拟，是一种异步通信方式，有一个特点：直接可以在用户态和系统态之间传递消息（一般用户态和系统态之间地址空间是隔离的，是不能够通讯的。就包括API函数以及shell命令，不能直接执行系统调用，必须使用API函数进一步调用）信号可以用户态和系统态之间直接传递消息（一个不同之处）

P244表 常见信号（必须知道）

SIGINT:敲键盘Ctrl-C

SIGQUIT:敲键盘Ctrl-\

这两个要记住，因为后面的信号集函数组处理的就是这两个信号，其他可以不知道

P248 信号处理函数

signal()函数、sigaction()函数的一般形式都是：

第一个参数发了一个信号，第二个参数就是信号处理函数，当它发出这个信号，立刻转由信号处理函数对它进行处理，有默认的处理方式，有自定义的处理方式，完全是由个人进行制定（知道一下信号处理就行）

P251信号集函数组

1. sigemptyset()
2. sigfillset()
3. sigaddset()
4. sigdelset()
5. sigismember()
6. sigprocmask()信号屏蔽字

6个信号处理函数

信号从创建、产生、注册、到屏蔽、解除屏蔽到生效整个流程全部包含其内，具体应用示例P252 sigset.c源文件 源程序自己看一下，相当于把这个流程再走一下

P254 信号量（重点考核）

再给大家提示一下，P255写的，最简单的信号量只能取0和1两种值，这种信号量被叫做二维信号量，在本节中，主要讨论二维信号量。而事实上，书上代码不是二维信号量，（为什么呢）你给一个信号量做一个V操作，它就溢出来了。V：加一操作

其次，信号量的生命周期与操作这个信号量的进程的生命周期是相同的

上课时做过演示，不同的状态的信号量的值不一样，体现出来它的生命周期和操作信号量的进程的生命周期是完全重合的

P258 两个源程序

左边sem\_com.c是有关信号量操作的函数库，右边是使用信号量函数库的调用的主程序（fork.c）

这两个源代码要考 考法：约束下谁先执行，谁后执行

你只要记得 ，既然这个信号量不是二维信号量，只有一个开关选项，当信号量初值为0的时候，不能执行P操作，对任何一个数都可以执行V操作，一直可以做下去

书上252的信号量，这个程序要考 考法，语句次序颠倒，问个问题，按照问题要求，实现这个功能（到时候需要把语句的顺序相应的调整下）

P260 共享内存

最为高效的进程间通信手段，因为不需要复制数据

本质原理：把内存当中的一个内存区域映射到两个内存的地址空间，使得两个进程读写自己的地址空间就相当于读写同一份共享内存的区域一样。

P262 共享内存源程序

网页中，作业要求程序运行结果，如何执行出来的，关键行号进行标注

因为这个程序不是用信号量做的，是用秘钥、字符串、root来做的，步骤过程比较多

把自己写的标注看看就行了，大概知道共享内存这个程序，它是做什么（就是子进程给父进程写数据，只不过写的时候，父进程读的时候，必须通过秘钥，才能够读取，否则不能读取）

P266 消息队列（了解下大概的功能）

一个进程给消息队列挂消息，另外一个进程从消息队列中取消息，first in first out ,操作永远是针对于队列的末尾的节点来执行（这个知道就行）

# 多线程

首先要明确一点，在编译带有多线程的源文件必须加一个选项参数( -p(不知道有没有空格)thread)

P292 使用信号量来约束线程和进程执行的次序

书上给的示例，create创建次序是012，但程序执行是210

网页作业要求完成120的执行次序

源代码要考，两种考法，一种：把201线程执行次序对应的代码应该做什么样的修正

另外一种（简单）：给出代码，分析执行次序

P298 生产者-消费者经典问题和信号量的综合解决方案（真正解决生产者和消费者问题，之前那个是仿真模拟，并不是真实的）

这个过程中，一共有三个实例，producer、customer、存放资源的区域（相当于资源库）

三个信号量，available（功能）表示有界缓冲区的空单元数，初始值为3，available这个信号量设置的意义是约束消费者的，意味着三个单元里的资源全部被customer消费掉了，再取的时候，available=0，没有可取资源。

空单元数为3，空单元数是三个单元，空表示没有资源，那么可用资源为0

full（功能）表示非空单元数，限制producer，生产者线程

这两个线程执行的先后次序，由mutex 信号量决定

书上初始值是1，可以把mutex初始值换成0

要求让生产者先执行，消费者后执行，直接把mutex信号量放到消费者线程的第一行，用一个wait的P操作来实现wait操作，P操作减一，初值为0，减一减不动，就意味着消费者进程被阻塞在P操作的位置，然后什么时候可以执行P操作呢？必须是生产者操作先执行，执行完以后最后来一个V操作，把mutex信号量的赋值由0改成1，改成1以后，消费者进程才能执行P操作，才能执行消费操作

（让消费者先执行没有意义，因为没有资源可消费，先执行也是要阻塞的）

生产者线程先执行，执行完以后V操作，赋值信号量由0改成1，消费者进程执行P操作

重点是把这三个信号量的功能，作用介绍了

程序要考，搞清楚这三个信号量意义、功能、价值，问的问题就能回答出来

一个次序，一个生产者与消费者的多线程、信号量的程序

# 网络编程

socket基础编程（效果）:客户端主动发起一个向服务端的socket链接，然后链接成功以后，客户端发起消息，服务端能收下，就行了。一次通信就结束了。（弊端/局限性：服务器只能接受客户端的一次消息传递，一次信息传递结束后就自己自动结束了。虽然监听队列可以一次监听5个，但是只要有一个客户端发个消息，被它接收下来，它只要一收下来，它立刻就结束了）

高级编程：让服务器可以同时接收多个客户端的消息的发送，此时就用到了select()函数，文件I/O编程中select()怎么用的这边就怎么用。（视频里有讲）

过程知道一下

基础编程和高级编程都是基于流式快捷字，基于TCP协议，所以工作流程图就是P313左边的流程图

代码太长，不考代码

工作原理，基于socket工具怎么实现通信的，工作流程，必须要掌握

工作流程：图10.6

图中所用函数：上面有介绍