1. ~~常用命令：ls, ps, cp, mv~~
2. ~~Shell语言：语法成分，shell和C可以相互调用，ps就是一个普通的app，path环境变量指向它，就执行了。命令行界面来自终端的shell程序的执行@实验6~~
3. ~~理解main，gcc是啥，components，not compiler，百宝箱，编译器工具箱，a.out是啥，改变进程图像，@实验5~~
4. ~~Fork(), 分裂，wait()等待进程运行结束和main函数返回值有关，八进制777，僵尸态，main函数返回值保存在pcb的处理机上下文里，iterator，父进程结束，应该会变成一号；~~
5. ~~改变（替换）进程图像（进程地址空间）那些数据，程序中所有内容都在地址空间里放着。Main函数里把它换了，进程并没有改变，执行的程序变了，进程和程序一一对应吗？不是！一个程序可以创建很多个进程，一个进程可以对应很多个程序，因为可以替换~~
6. ~~Shell语言~~
7. ~~管道：进程之间加管道，c语言写的可执行文件只是软件流水线上的一个节点，那些东西都是零件，一个命令就做一件事，把它装配起来。怎么把前一个进程的printf送到后一个进程的scanf。没有必要遍历，加个ls，加个sort之类~~
8. ~~信号使用，进程之间如何相互控制，怎么把另外一个干掉。A. 怎么进入调试状态，靠中断，指令指针和某个值对上了，引起异常，断点就和断点寄存器指针对上，就挂起，受IDE控制，受信号控制，ptues，使进程停下、改变量、继续进行….变量在os里对应地址，-g—符号表，原理就是信号。千万不要读死书，各个进程之间不能相互影响，都是独立的balabala，。~~
9. 存储映射文件mmap，理解他就能理解虚拟存储，很老了，1987年。进程的地址空间某一段对应外存的一个文件，建立这样一个关系。进程运行空间分为几个段，每一段对应一个文件。存储扩充还是文件缓存？所有数据都是文件数据，读，文件缺页中断，写，页面置换算法。虚拟存储存储到哪里？存储到文件，文件是磁盘上构建的一个抽象

假设有多个磁盘请求，先执行哪个？选择一个，不是并发读

磁盘存在于设备中，文件系统和设备是无关的

微内核