在一个程序中声明了却没有定义和调用的函数不会出现在这个程序的符号表中

在一个程序中调用了的函数会出现在这个程序的符号表中

在一个程序中引用了的函数或变量都会出现在这个程序的符号表中，无论其是否在该程序中进行过声明或定义

一个没有初始化的全局数组名是弱符号

一个没有初始化的全局指针是弱符号

已初始化的全局变量、定义了函数体的函数是强符号

未初始化的全局变量、出现在符号表却没有定义函数体的函数是弱符号

在进行符号解析时优先使用局部符号

所有函数声明自带extern，而变量声明不带

有extern标识的符号如果没有在程序中被引用那么不会出现在这个程序的符号表中

有const修饰的全局变量、printf的格式串、switch语句的跳转表会被放在.rodata节中

有static修饰的函数或变量都是局部的

已初始化不为零的全局变量或静态局部变量会被放在.data节中

已初始化为0的全局变量和所有初始化为0或未初始化的静态局部变量会被放在.bss节中，运行时直接分配0

未初始化的全局变量会被放在COMMON伪节中

所有由extern修饰的外部符号会被分配在UNDEF伪节中

调用外部函数或引用全局变量的指令需要在.rel.text节中有重定位信息

已初始化的全局变量如果初始值为一个全局变量的地址或者外部定义函数的地址，那么都需要在.rel.data中有重定位信息

动态链接共享库以.so为结尾，动态链接器本身就是一个共享目标（如ld-linux.so），共享库编译必须总是使用位置无关代码的技术，使用命令-fpic

一般保护故障是由于一个程序引用了未定义的虚拟内存区域或试图写一个只读的文本段造成的（比如页表中标记为只读的部分），也就是SEGV

SIGKILL和SIGSTOP既不能被捕获，也不能被忽略

setjmp在env缓冲区中保存当前调用环境，以便后面longjmp使用，而longjmp从env缓冲区中恢复调用环境，然后触发一个从最近一次初始化env的setjmp的调用的返回，然后setjmp返回，并以longjmp的第二个参数为返回值，setjmp调用一次但可能返回多次，而longjmp调用一次并从不返回，可能会导致潜在的内存泄漏（跳过了释放内存的语句），sigsetjmp和siglongjmp不是异步信号安全的

DRAM采用写回而不是直写，层次较低的高速缓存更注重命中率，层次较高的高速缓存更注重命中时间，在不命中处罚低的高速缓存上使用较低的相联度，在不命中处罚高的高速缓存上使用较高的相联度，DRAM缓存是全相联的。

只有一级页表需要常驻主存，只有最经常使用的二级页表才需要缓存在主存中

Corei i7使用四级页表，每一级页表包含40位的地址（这样物理页表和物理页都是4KB对齐的），第0位表示子页表是否在物理内存中，第1位表示读写权限，第二位表示用户模式或内核模式的权限，第三位表示直写或写回策略，第四位表示能不能缓存，第五位表示引用位，第六位只在第四级页表中使用，表示是否被修改（dirty），七位表示页大小4K还是4M（只对第一级页表有效），最高位为XD位表示是否可执行

一个对象可以被映射到虚拟内存的一个区域，如果作为共享对象那么这个进程对这个区域的写操作都对共享这个对象的进程可见并会反应在原始对象当中

而如果作为私有区域，任何修改对其它进城不可见并且对这个区域的写操作不会反应在这个对象上

试图写一个私有区域的页面会触发保护故障，进行写时复制copy on write

可以用stat和fstat读取文件元数据，stat以文件名为输入，fstat以文件描述符为输入

每个进程有独立的描述符表，所有进程共享打开文件表和vnode表，dup2会把第二个参数重定向到第一个参数

以太网（局域网LAN）——（网桥）——桥接以太网——路由器——广域网（WAN）

TCP/IP协议运行在每个因特网主机上，

IP协议提供基本的命名方法和递送机制，向其他主机发送包或数据报，不可靠

UDP扩展了IP协议，包可以在进程间而不是主机间传送，不可靠

TCP构建在IP之上，是进程间可靠且全双工的链接

HTTP超文本传输协议在Web客户端和服务器间交互，Web内容使用HTML超文本标记语言编写

对Web客户端和服务器而言内容是与MIME（多用途网际邮件扩充协议）类型相关的字节序列

HTTP请求有一个请求行后面跟随若干个请求报头和一个空的文本行组成，一个请求行的格式为method URI version，一般method=GET，直到服务器生成和返回URI（统一资源标识符）标识的内容，其是相应的URL的后缀，version表明了请求遵循的http版本