Lesson 1

1.程序从编写到运行：

a) Precompile：主要是处理了以#开头的指令（预编译指令），#include　#ifndef　#define

b) Compile　编译针对的是单个源文件，生成.o 目标文件

c) Link 将每个目标文件链接起来，生成可执行文件

2.头文件中不加上预编译指令，造成的重复定义是编译期错误。

3.头文件中定义 int a = 10;之类的语句属于链接期间的错误。

4.左值和右值：左值就是可以放在表达式左边或者右边的值，右值则只能放在表达式的右边。

5.指针的两重**语义**（semantics)：

a) 所指向内存空间的 base 地址

b) 本身的类型信息

6.所有的变量都可以取地址（&），所有的指针除了 void *之外都可以进行解引用(dereference)，void*本身不包含类型信息。

7.交换 T 类型变量，采用指针的办法是：

```
void swap(T *a, T *b){
    T temp = *a;
    *a = *b;
    *b = temp;
}
```

8.const int *p = &i 这里仅仅是不能通过 p 修改 i，例如*p = 2

是错误的，但是 i 的值可以任意修改，i = 34 是正确的。

9.关于数组传参：

a) 一维数组用 int *做参数。在数组退化成 int *的过程中，数组仍可以正常使用，但是丢失了长度信息。

b) 二维数组 int A[4][3]，采用 int (*)[3]作为参数，int (*)[3]是一个数组指针。这里丢失了第一维，所以需要手工指定第一维的长度。

10.解决互斥问题的方案是：

a) 互斥锁、信号量

b) 原子操作，gcc 提供 CAS

11.互斥是一种竞争关系，同步则是一种合作关系。

12.有可能产生互斥问题的场景称为竞态条件。

13.Linux 中创建进程的方式有：

a) fork

b) Vfork

c) clone

14.fork 采用了一种“写时复制”(copy on write)的技术。在传统的 UNIX 的进程模型中，fork 进程是把整个进程的项复制一份，开销巨大。COW 就是在以前的基础上，子进程仅仅复制父进程的页表，然后设为只读，任何一方试图修改项，就将该项单独复制一份。

15.Clone 用于创建线程。在 Linux 中，线程是采用了一种轻

量级进程的实现。所以 Linux 中的线程有两个 ID，pthread_self()是 pthread_t 类型是所谓的线程 id，然而还有一个真实的进程 ID，类型 pid_t，获取方式为 gettid().

16.我们采用的线程模型叫做 NPTL，（native posix thread)，它采用的是 1:1 的线程模型。