<<从Windows到Linux的应用移>>

图书基本信息

书名：<<从Windows到Linux的应用移植实现>>

13位ISBN编号：9787118082524

10位ISBN编号：711808252X

出版时间：2013-1

出版时间：国防工业出版社

作者：兰雨晴，洪雪玉 著

页数：240

<<从Windows到Linux的应用移>>

内容概要

应用系统的开发、运行与部署依赖于所选择平台的工具、语言、环境、开发技术、服务及其兼容的第三方软件。
因此进行移植时，需要考虑目标平台对上述因素的支持和实现情况。
具体地，可以从应用系统组件、用户工作环境、开发工具／语言／平台、应用开发技术等几个角度整体考虑。
《从windows到linux的应用移植实现：平台技术与接口篇》主要关注应用开发技术在Windows平台和Linux平台的实现情况，研究分析网络通信、多进程侈线程、图形界面等技术在两个平台的实现差异性，并提供对应的实现方案。

在内容编排上，为了让读者对应用移植涉及的工作和过程有清晰的认识，《从windows到linux的应用移植实现：平台技术与接口篇》在第一部分定义了应用移植过程、主要阶段、各阶段主要工作、应用移植的几个角度等，然后分为不同的部分，介绍多项应用开发技术的移植实现。
其中，第二部分从第2章到第5章，介绍基于套接字网络通信技术的应用移植实现；第三部分从第6章到第9章，介绍基于并发多任务开发技术的应用移植实现；第四部分从第10章到第12章，介绍图形用户界面的应用移植实现。

<<从Windows到Linux的应用移>>

书籍目录

章节摘录

版权页： 插图： 2.Linux对应接口 Linux共享内存的创建/打开操作相比Windows要简单一些，System V使用shmget（）函数来创建新的或取得已有的共享内存。
它在Linux系统库linux/shm.h中的定义是这样的： #include #include int shmget（key_t key，int size，int shmflg）；参数key指定了共享内存的关键字。
第二个参数size是创建的共享内存的大小，以字节为单位。
第三个参数shmfl9是控制函数行为的标志量。
如果操作成功，函数返回共享内存的标识符。
而POSlX中调用函数shin_open（）g建/打开共享内存。
函数原型为： #include #include #include int shm_open（const char name，int oflag，mode—t mode）；参数name指定了将要创建/打开的共享内存对象。
参数0fla9用来指定访问标识、创建标识和状态标识等，取值如下：O_RDONLY、O_RDWR
、0_CREAT、O_EXCL、O_TRUNC。
mode指定了该共享内存的权限。
System V关闭共享内存是通过调用函数shmctl（）来实现的，它在Linux系统函数库include/linux/shm.h中的函数声明是这样的： #include #include int shmctl（int shmqid，int cmd，struct shmid—ds buf）；其中，参数shmqid是shmget（）函数返回的共享内存标识符。
第二个参数cmd是要执行的操作，可取的值有：IPC_STAT，表示将buf指向的shmid_ds结构中的数据设置为共享内存的当前关联值；IPC_SET，表示如果进程有足够的权限，就把共享内存当前关联的值设置为buf指向的结构中的值；IPC_RMID，表示删除共享内存。
参数buf指向一个包含共享内存模式和访问权限的结构。
需要说明的是，当执行IPC_RMID操作时，系统并不是立即将其删除，而只是将其标为待删，然后等待与其连接的进程断开连接。
只有当所有的连接都断开以后系统才执行真正的删除操作。
当然，如果执行IPC_RMID的时候没有任何的连接，删除操作将会被立即执行。
POSIX调用函数shm unlink（）关闭共享内存。
函数原型为： #include #include #include int shm unlink（const char name）；该函数类似于unlink（），它根据参数name删除指定共享内存对象的连接；当所有进程都删除了该共享内存对象的连接时，系统将删除共享内存并释放相应的内存空间。

<<从Windows到Linux的应用移>>

编辑推荐

《从Windows到Linux的应用移植实现:平台技术与接口篇》主要关注应用开发技术在Windows平台和Linux平台的实现情况，研究分析网络通信、多进程侈线程、图形界面等技术在两个平台的实现差异性，并提供对应的实现方案。

<<从Windows到Linux的应用移>>

版权说明