

操作系统上机实验题目

实验 1

实验题目

创建进程

实验目的

学会通过基本的 Windows 或者 Linux 进程控制函数，由父进程创建子进程，并实现父子进程协同工作。

实验软件环境

VC++6.0 或 Linux 操作系统

实验内容

创建两个进程，让子进程读取一个文件，父进程等待子进程读取完文件后继续执行，实现进程协同工作。

进程协同工作就是协调好两个进程，使之安排好先后次序并以此执行，可以用等待函数来实现这一点。当需要等待子进程运行结束时，可在父进程中调用等待函数。

实验 2

实验题目

线程共享进程数据

实验目的

了解线程与进程之间的数据共享关系。创建一个线程，在线程中更改进程中的数据。

实验软件环境

VC++6.0 或者 Linux 操作系统

实验内容

在进程中定义全局共享数据，在线程中直接引用该数据进行更改并输出该数据。

实验 3

实验题目

信号通信

实验目的

利用信号通信机制在父子进程及兄弟进程间进行通信。

实验软件环境

VC++6.0 或者 Linux 操作系统

实验内容

父进程创建一个有名事件，由子进程发送事件信号，父进程获取事件信号后进行相应的处理。

实验 4

实验题目

匿名管道通信

实验目的

学习使用匿名管道在两个进程间建立通信。

实验软件环境

VC++6.0 或者 Linux 操作系统

实验内容

分别建立名为 Parent 的单文档应用程序和 Child 的单文档应用程序作为父子进程，由父进程创建一个匿名管道，实现父子进程向匿名管道写入和读取数据。

实验 5

实验题目

命名匿名管道通信

实验目的

学习使用命名匿名管道在多进程间建立通信。

实验软件环境

VC++6.0 或者 Linux 操作系统

实验内容

建立父子进程，由父进程创建一个命名匿名管道，由子进程向命名管道写入数据，由父进程从命名管道读取数据。

实验 6

实验题目

信号量实现进程同步

实验目的

进程同步是操作系统多进程/多线程并发执行的关键之一，进程同步是并发进程为了完成共同任务采用某个条件来协调他们的活动，这是进程之间发生的一种直接制约关系。本次试验是利用信号量进行进程同步。

实验软件环境

VC++或者 Linux 操作系统

实验内容

- 生产者进程生产产品，消费者进程消费产品。
- 当生产者进程生产产品时，如果没有空缓冲区可用，那么生产者进程必须等待消费者进程释放出一个缓冲区。
- 当消费者进程消费产品时，如果缓冲区中没有产品，那么消费者进程将被阻塞，直到新的产品被生产出来。

实验 7

实验题目

共享主存实现进程通信

实验目的

利用共享主存解决读写者问题。要求写者进程创建一个共享主存，并向其中写入数据，读者进程随后从该共享主存区中访问数据。

实验软件环境

VC++或者 Linux 操作系统

实验内容

为基于共享主存解决读者-写者问题，需要由写进程首先创建一个共享主存，并将该共享主存区映射到虚拟地址空间，随后读进程打开共享主存，并将该共享主存区映射到自己的虚拟地址空间，从中获取数据，并进行处理，以此实现进程通信。

实验 8

实验题目：

模拟实现一个 Linux 文件系统

实验目的：

熟悉 Linux 文件系统。熟悉文件系统的管理，掌握成组链接法等方法。

实验软件环境：

Linux 操作系统或者 Windows 中 VC++

实验内容：

在磁盘空间模拟实现一个 Linux 文件系统，并提供基本的文件操作命令（如 mk, cp, mkdir, rmdir, cd, ls, cat, chmod, chown, chgrp, chnam 等）。文件系统的实现要求采用混合索引式文件结构，包括使用空闲节点号栈管理空闲节点和成族链接管理空闲盘块；

实验 9

实验题目：

通过内核模式显示进程控制块信息

实验目的：

了解 Linux 内核模块变成，熟悉 Linux 内核机制。

实验软件环境：

Linux 操作系统

实验内容：

在内核中，所有的进程控制块都被一个双向链表连接起来，该链表中的第一个进程控制块为 init_task。编写一个内核模块，模块接收用户传递的一个参数 num， num 指定要打印的进程控制块的数量；若用户不指定 num 或者 num<0，模块则打印所有进程控制块的

信息，需要打印的进程控制块信息由：进程 PID 和进程的可执行文件名

实验 10（选作）

实验题目：

WRK 内存管理实验

实验目的：

了解 Windows 内存管理策略及基本数据结构

理解 Windows 的地址过程

实验软件环境：

Windows 平台下 Windbg 调试工具

实验内容：

Windows 的地址转译过程，任意给出一个虚拟地址，通过 WinDbg 观察相关数并找到其物理地址