

第11讲 操作系统如何控制 进程的并发执行



Typical Function of an OS Kernel (资源管理功能)

- **Process Management** : 进程创建和终止、调度、状态转换、同步和通信、管理 PCB
- **Memory Management** : 为进程分配地址空间、对换、段 / 页管理
- **I/O Management** : 缓存管理、为进程分配 I/O 通道和设备



Typical Function of an OS Kernel(支撑功能)

- Interrupt handling (中断处理)
- Timing (时钟管理)
- Primitive (原语) : Atomic Operation
- Accounting (统计)
- Monitoring (监测)



Process Control Primitives (原语)

- Process Switch , 进程切换
- Create and Terminate, 创建与终止
- Block and Wakeup, 阻塞与唤醒
- Suspend and Activate, 挂起与激活



When to Switch a Process

- **Clock interrupt**

- process has executed for the maximum allowable time slice （时间片）

- **I/O interrupt**

- **Memory fault （存储访问失效）**

- memory address is in virtual memory so it must be brought into main memory



When to Switch a Process

- **Trap (陷阱)**
 - error occurred
 - may cause process to be moved to Exit state
- **Supervisor call (管理程序调入)**
 - such as file open



Change of Process State

- **Save context** of processor including program counter and other registers
- Update the PCB of the process that is currently running
- Move PCB to appropriate queue - ready, blocked
- Select another process for execution



Change of Process State

- Update the PCB of the process selected
- Update memory-management data structures
- Restore context of the selected process

Process Switching vs. Mode Switching

- **Process Switch**, 是作用于进程之间的一种操作。当分派程序收回当前进程的 CPU 并准备把它分派给某个就绪进程时, 该操作将被引用。
- **Mode Switch**, 是进程内部所引用的一种操作。当进程映像所包含的程序引用核心子系统所提供的系统调用时, 该操作将被引用。

Process Creation

- Submission of a batch job
- User logs on
- Created to provide a service such as printing
- Process creates another process



Process Creation (creat() 原语, 步骤)

1. 为进程分配一个唯一标识号 ID: 主进程表中增加一个新的表项
2. 为进程分配空间 : 用户地址空间、用户栈空间、PCB 空间。若共享已有空间, 则应建立相应的链接。
3. 初始化 PCB: 进程标识、处理机状态信息、进程状态
4. 建立链接 : 若调度队列是链表, 则将新进程插入到就绪或 (就绪, 挂起) 链表
5. 建立或扩展其他数据结构

Process Termination(终止)

- Batch job issues *Halt* instruction
- User logs off
- Quit (退出) an application
- Error and fault conditions

Reasons for Process Termination

- Normal completion, 正常结束
- Time limit exceeded, 超时终止, 执行时间超过预计时间
- Memory unavailable, 内存不足, 无法为进程分配所需的内存空间
- Bounds violation, 越界访问
- Protection error, 企图使用未允许用的数据, 或操作方式错
- Arithmetic error, 计算错, 如除零, 或企图存储硬件允许的最大数
- Time overrun, 超时等待某事件发生

Reasons for Process Termination

- I/O failure, 如找不到文件或多次重试仍无法读写文件, 或无效操作
- Invalid instruction, 企图执行不存在的指令
- Privileged instruction, 企图执行特权指令
- Data misuse, 数据类型不符, 或未初始化
- Operating system intervention, 操作员或 OS 干预, 如发生死锁的时候
- Parent terminates, so child processes terminate
- Parent request

Process Termination (destroy() 原语, 步骤)

1. 根据被终止进程的标识符 ID , 找到其 PCB , 读出该进程的状态;
2. 若该进程为执行状态, 则终止其执行, 调度新进程执行;
3. 若该进程有子孙进程, 则立即终止其所有子孙进程
4. 将该进程的全部资源, 或归还给其父进程, 或归还给系统
5. 将被终止进程 (的 PCB) 从所在的队列中移出, 等待其它程序来搜集信息



Process Block and Wakeup (阻塞的原因)

- 请求系统服务
- 启动某种操作：如 I/O
- 新数据尚未到达
- 无新工作可做



Process Block and Wakeup (原语)

- 阻塞原语 `block()`

当出现阻塞事件，进程调用阻塞原语将自己阻塞。状态变为“阻塞状态”，并进入相应事件的阻塞队列

- 唤醒原语 `wakeup()`

当阻塞进程期待的事件发生，有关进程调用唤醒原语，将等待该事件的进程唤醒。状态变为 Ready，插入就绪队列

Process Block and Wakeup (原语)

- 挂起原语 `suspend()`

当出现挂起事件，系统利用挂起原语将指定进程或阻塞状态进程挂起。进程从内存换到外存，状态改变：
`Ready Ready,Suspend; Blocked Blocked,Suspend` ,
插入相应队列

- 激活原语 `active()`

当激活事件发生，系统利用激活原语将指定进程激活。进程从外存换入到内存，状态改变：
`Ready,Suspend Ready ;Blocked,Suspend Blocked` ,
插入相应队列

