Processes Management

- GU/Jianhua
- School of Computer Science
- Northwestern Polytechnical University

Processes Management

- Process Concept
- Threads
- CPU Scheduling
- Process Synchronization
- Deadlocks

Process Concept

- Process Description
  - Process and Features
  - Process State
  - Process Control Block (PCB)
- Process Control

Process Concept(1)

- Program and Process
  - Multi-programmed:
    - 程序的运行结果与它们的相对速度有关。
    - 程序与它的执行过程不再一一对应。
    - 并发程序之间存在直接或间接的依赖和制约关系。
- Process Concept: a program in execution
  - 进程是程序及其数据在计算机上的一次运行，是系统进行调度和资源分配的独立单位

并发进程演示

Process features

- **动态性**：进程是程序的一次执行，它有着“创建”、“活动”、“暂停”、“撤消”等过程，具有一定的生命期，是动态地产生、变化和消亡的。
- **并发性**：进程之间的动作在时间上可以重叠，即系统中有若干进程都已经“开始”但又没有“结果”，称这些进程为并发进程。
- **独立性**：进程是系统调度和资源分配的独立单位，它具有相对独立的功能，拥有自己独立的进程控制块PCB。
- **异步性**：各个并发进程按照各自独立的、不可预知的速度向前推进。
- **交互性**：由于并发进程之间具有直接或间接的关系，在运行过程中它们之间需要进行必要的交互（同步、互斥和数据通信等），以完成特定的任务。

Program and Process

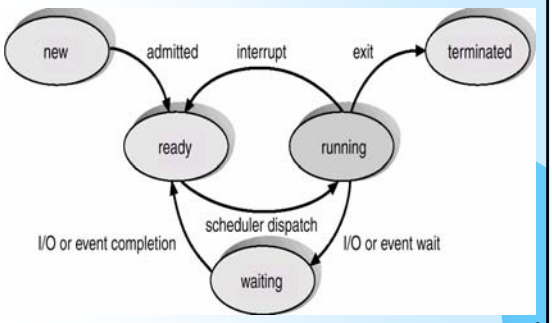
- 进程是程序及其数据在计算机上的一次运行活动，它属于一种动态的概念(Active entity)。进程的运行实体是程序，离开程序进程没有存在的意义。从静态角度看，进程是由程序、数据和进程控制块PCB三部分组成的。而程序是一组有序的指令集合，属于一种静态的概念(Passive entity)。
- 进程是程序的一次执行过程，它是动态地创建和消亡的，具有一定的生命期，是暂时存在的；而程序则是永久存在的，可长期保存
- 一个进程可以执行一个或几个程序，一个程序也可以构成多个进程。

Process State

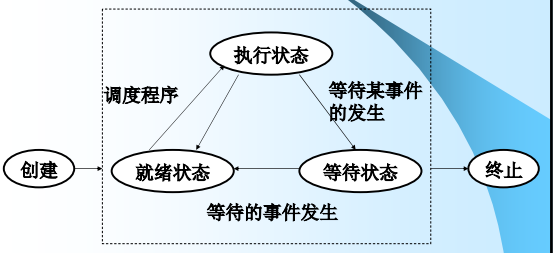
Process State:

- **new**: The process is being created.
- **running**: it holds the CPU and is executing instructions
- **waiting(blocked)**: it is waiting for some event to occur.
- **ready**: The process is waiting to be assigned to a process.
- **terminated**: The process has finished execution.

Diagram of Process State



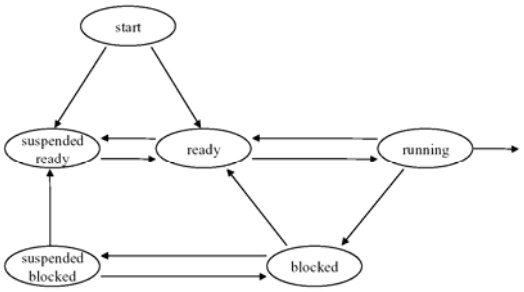
进程状态变化图



Suspended Processes(1)

- **Suspend state**:
  - When all of the processes in main memory are in the blocked state, the operating system can suspend one process by putting it in the **Suspend state** and transferring it to disk. The space that is freed in main memory can then be used to bring in another process.
- **Blocked, suspend:**
  - The process is in secondary memory and awaiting an event.
- **Ready, suspend:**
  - The process is in secondary memory but it is available for execution as soon as it is loaded into main memory.

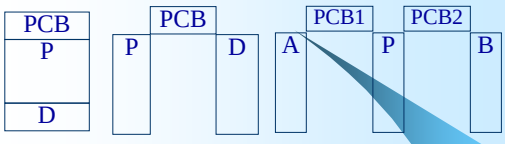
Suspended Processes (2)



Process Control Block (PCB)

- **A process includes:**
  - Program: text section
  - program counter
  - stack
  - data section
- **Role of PCB**
  - One process, one PCB
  - Trace the information specific to a process
  - Manage process
- **Process**: program+data+PCB

Pure Code



- 所谓纯代码，又称为可再入码（reentrant code），是可以为多个进程共享的程序段，代码不因程序的执行而改变。

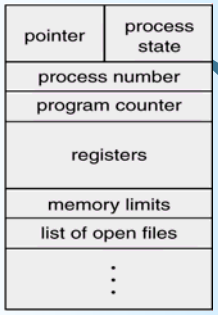
Elements of a PCB(1)

- process identification:
  - process id: numeral OR string
  - parent process id
  - user id
- processor state information:
  - register set
  - program counter
  - stack counter
  - condition codes
  - processor status

Elements of a PCB(2)

- process control information:
  - process state
  - scheduling information
  - event (wait-for)
  - memory-mgmt info
  - resources

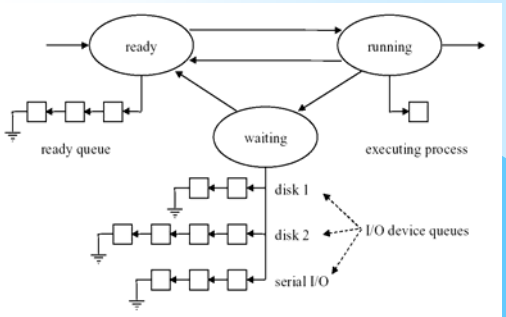
An Example for PCB



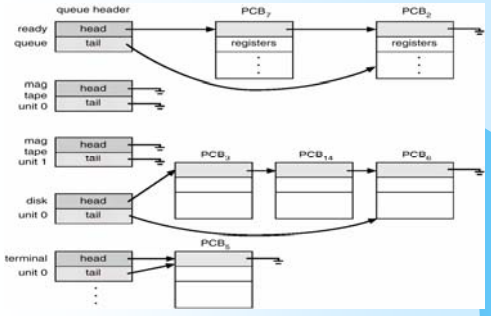
Organization of PCB

- Linear List
- Linked List
  - **Job queue** – set of all processes in the system.
  - **Ready queue** – set of all processes residing in main memory, ready and waiting to execute.
  - **Device queues** – set of processes waiting for an I/O device. Also called *waiting queue*.
  - Process migration between the various queues.

Linked List



Ready Queue And Various I/O Device Queues



Process Concept

- Process Description
  - Process and Features
  - Process State
  - Process Control Block (PCB)
- Process Control

Process Control

- Process Creation
- Process Termination
- Process Waiting
- Process Waking up

Primitive

- 所谓“原语”是计算机机器指令的延伸，它是由若干条机器指令构成、完成一种特定功能的程序段；为保证操作的正确性，还规定原语在执行期间必须一次执行完，中间不允许被中断。
- 原语具有**原子性**。
- 执行原语的过程中一般要关中断
- 原语举例：银行转账

Process Creation(1)

- Parent process creates children processes, which, in turn create other processes, forming a tree of processes.
- Resource sharing
  - Parent and children share all resources.
  - Children share subset of parent's resources.
  - Parent and child share no resources.
- Execution
  - Parent and children execute concurrently.
  - Parent waits until children terminate.

进程家族树的优点

- 资源分配严格，子进程只能继承父进程所拥有的资源，便于管理；
- 系统可根据需要赋予进程不同的控制权，并可以把一个任务分解成若干个进程来完成，具有较好的灵活性；
- 树形结构层次清晰，关系明确。

Process Creation (2)

- Address space
  - Child duplicate of parent.
  - Child has a program loaded into it.
- UNIX examples
  - **fork()** system call creates new process
  - **execve()** system call used after a **fork** to replace the process' memory space with a new program.

Process Creation (3)

- When a process created?
  - Submission of a batch job
  - User logs on
  - Create process to provide service such as printing
  - Spawned by existing processes

进程创建的基本过程

- 首先从空闲的PCB集合中申请一个新的PCB，同时获得该进程的内部标识；
- 然后向该PCB中填写各种参数；
- 把该进程的状态设置成就绪状态，并将该PCB插入到就绪队列中。

Process Termination（1）

- 撤消进程的两种策略：
  - 撤消指定进程（包括调用进程）
  - 撤消该进程及其所有子孙进程

Process Termination（2）

- Process executes last statement and asks the operating system to decide it (**exit**).
  - Output data from child to parent (via **wait**).
  - Process' resources are de-allocated by operating system.
- Parent may terminate execution of children processes (**abort**).
  - Child has exceeded allocated resources.
  - Task assigned to child is no longer required.
  - Parent is exiting.
    - Operating system does not allow child to continue if its parent terminates.
    - Cascading termination.（a process and all its children)

进程终止的基本过程

- 找到相应进程的PCB；
- 若进程正处于执行状态，则立即停止，设置重新调度标志，必要时撤消属于该进程的所有"子孙"进程，释放被撤消进程的所有资源、释放进程的PCB；
- 若调度标志为真，则进行重新调度

Process Waiting

- 使调用该原语的进程变为等待状态；
- 将指定的进程变为等待状态；
- 将某进程及其所有子孙进程变为等待状态。

进程等待的基本过程

- 找到相应进程的PCB;
- 如果该进程为执行状态，则保护其现场，将其状态改变为等待状态，停止运行，并把该PCB插入到相应的等待队列中去;
- 若为就绪状态，则将其状态修改为等待状态，把它移出就绪队列，并插入到等待队列中去

Process Waking up

- 进程因等待某事件的发生而处于等待状态，当等待事件发生后，就要用唤醒原语将其唤醒。
- 唤醒原语的基本操作是：在等待队列中找到相应进程的PCB，将其从等待队列中移出，并置其状态为就绪状态，然后把该PCB插入就绪队列中，等待调度程序调度。

Unix Process and Control

- Unix *proc* and *user* structure
- Unix Process State
- Unix Process Control
  - Creation
  - Termination
  - Waiting