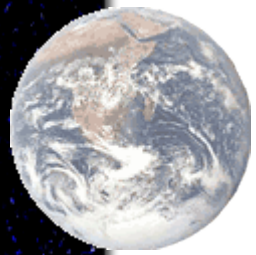
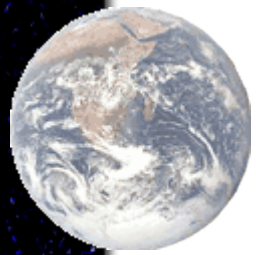


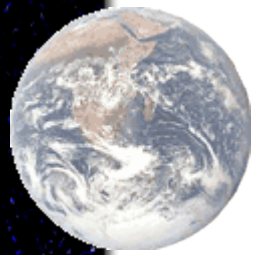
Operating System Control Structures

- Information about the current status of each process and resource
- Tables are constructed for each entity the operating system manages



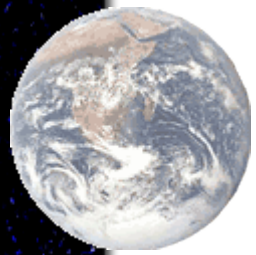
Memory Tables

- Allocation of main memory to processes
- Allocation of secondary memory to processes
- Protection attributes for access to shared memory regions
- Information needed to manage virtual memory



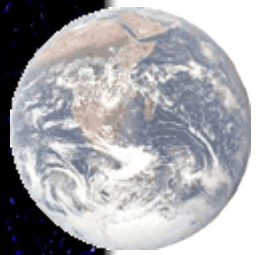
I/O Tables

- I/O device is available or assigned
- Status of I/O operation
- Location in main memory being used as the source or destination of the I/O transfer



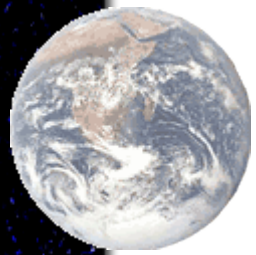
File Tables

- Existence of files
- Location on secondary memory
- Current Status
- Attributes
- Sometimes this information is maintained by a file management system



Process Table

- Where process is located
- Attributes in the process control block
  - Program
  - Data
  - Stack



Process Image

Table 3.4 Typical Elements of a Process Image

User Data

The modifiable part of the user space. May include program data, a user stack area, and programs that may be modified.

User Program

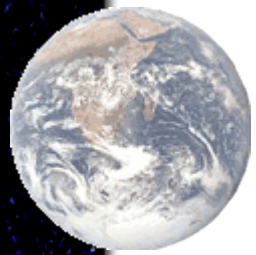
The program to be executed.

System Stack

Each process has one or more last-in-first-out (LIFO) system stacks associated with it. A stack is used to store parameters and calling addresses for procedure and system calls.

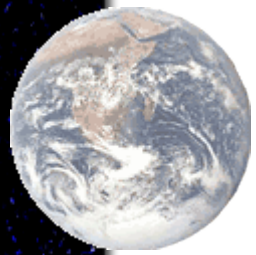
Process Control Block

Data needed by the operating system to control the process (see Table 3.5).



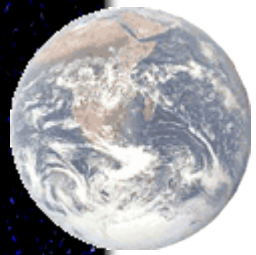
Process Control Block

- Process identification
  - Identifiers
    - Numeric identifiers that may be stored with the process control block include
      - Identifier of this process
      - Identifier of the process that created this process (parent process)
      - User identifier



Process Control Block

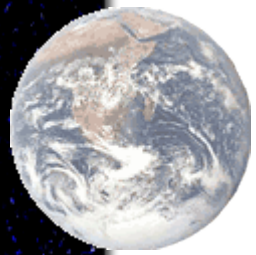
- Processor State Information
  - User-Visible Registers
    - A user-visible register is one that may be referenced by means of the machine language that the processor executes while in user mode. Typically, there are from 8 to 32 of these registers, although some RISC implementations have over 100.



Process Control Block

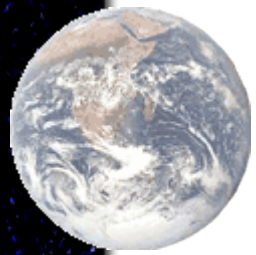
- Processor State Information
  - Control and Status Registers

    These are a variety of processor registers that are employed to control the operation of the processor. These include
    - *Program counter:* Contains the address of the next instruction to be fetched
    - *Condition codes:* Result of the most recent arithmetic or logical operation (e.g., sign, zero, carry, equal, overflow)
    - *Status information:* Includes interrupt enabled/disabled flags, execution mode



Process Control Block

- Processor State Information
  - Stack Pointers
    - Each process has one or more last-in-first-out (LIFO) system stacks associated with it. A stack is used to store parameters and calling addresses for procedure and system calls. The stack pointer points to the top of the stack.



Process Control Block

- Process Control Information
  - Scheduling and State Information

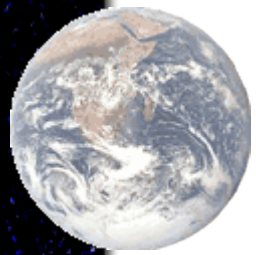
    This is information that is needed by the operating system to perform its scheduling function. Typical items of information:

    •*Process state:* e.g., running, ready, waiting, halted

    •*Priority:* scheduling priority of the process

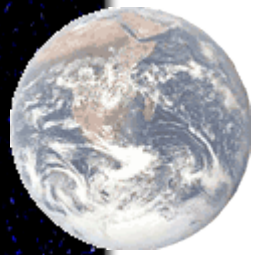
    •*Scheduling-related information:* e.g. waiting time, last run execution amount of time and so on… (depends on the algorithm)

    •*Event:* Identity of event the process is awaiting before it can be resumed



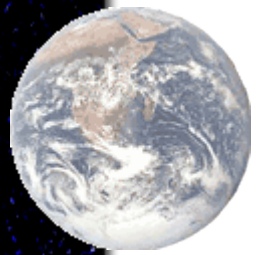
Process Control Block

- Process Control Information
  - Data Structuring
    - A process may be linked to other process in a queue, ring, or some other structure. For example, all processes in a waiting state for a particular priority level may be linked in a queue. A process may exhibit a parent-child (creator-created) relationship with another process. The process control block may contain pointers to other processes to support these structures.



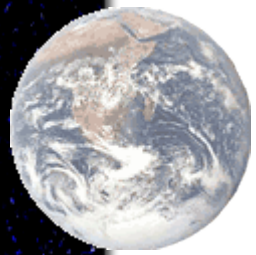
Process Control Block

- Process Control Information
  - Interprocess Communication
    - Various flags, signals, and messages may be associated with communication between two independent processes. Some or all of this information may be maintained in the process control block.
  - Process Privileges
    - Processes are granted privileges in terms of the memory that may be accessed and the types of instructions that may be executed. In addition, privileges may apply to the use of system utilities and services.



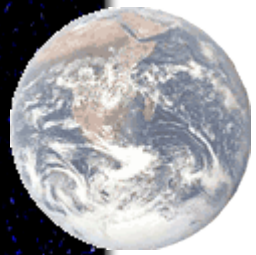
Process Control Block

- Process Control Information
  - Memory Management
    - This section may include pointers to segment and/or page tables that describe the virtual memory assigned to this process.
  - Resource Ownership and Utilization
    - Resources controlled by the process may be indicated, such as opened files. A history of utilization of the processor or other resources may also be included; this information may be needed by the scheduler.



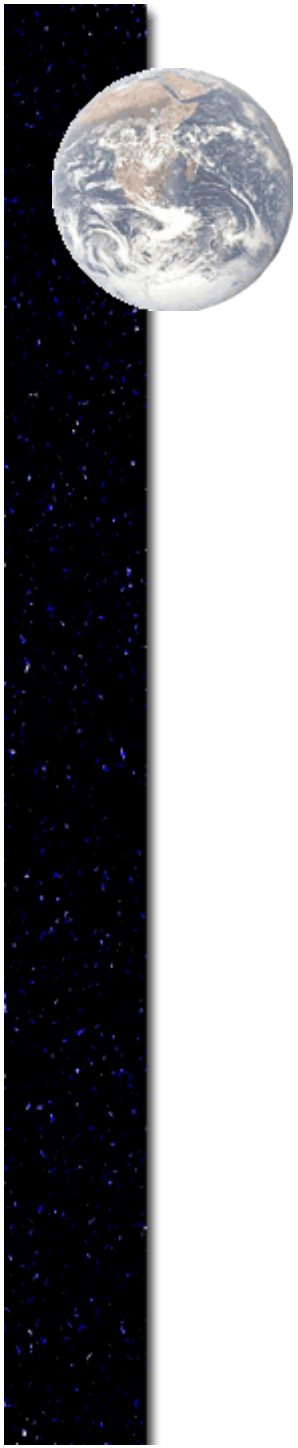
Modes of Execution

- User mode
  - Less-privileged mode
  - User programs typically execute in this mode
- System mode, control mode, or kernel mode
  - More-privileged mode
  - Kernel of the operating system



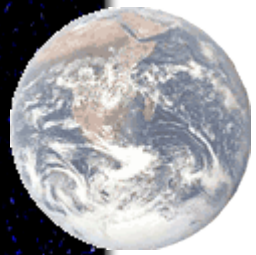
Changing execution mode

- When:
  - User Program issues call to OS service
  - Interrupt triggers exec OS routine
  - When returning from OS routine to user process
- PSW- bit for execution mode
  - Lower-privileged mode, higher value
  - Kernel mode typically 0



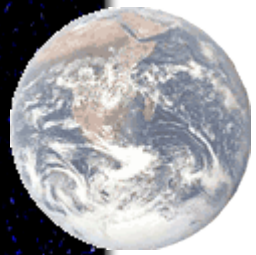
Process Creation

- Assign a unique process identifier
- Allocate space for the process
- Initialize process control block
- Set up appropriate linkages
  - Ex: add new process to scheduling queue (ready OR ready/suspend)
- Create of expand other data structures
  - Ex: maintain an accounting file



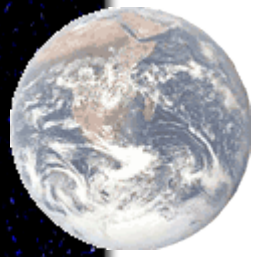
When to Switch a Process

- Clock interrupt
  - process has executed for the maximum allowable time slice
- I/O interrupt
- Memory fault
  - memory address is in virtual memory so it must be brought into main memory



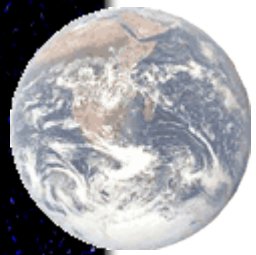
When to Switch a Process

- Trap
  - error or exception occurred
  - may cause process to be moved to Exit state
- Supervisor call
  - such as file open



Change of Process State

- Save context of processor including program counter and other registers
- Update the process control block of the process that is currently in the Running state
- Move process control block to appropriate queue – ready; blocked; ready/suspend
- Select another process for execution

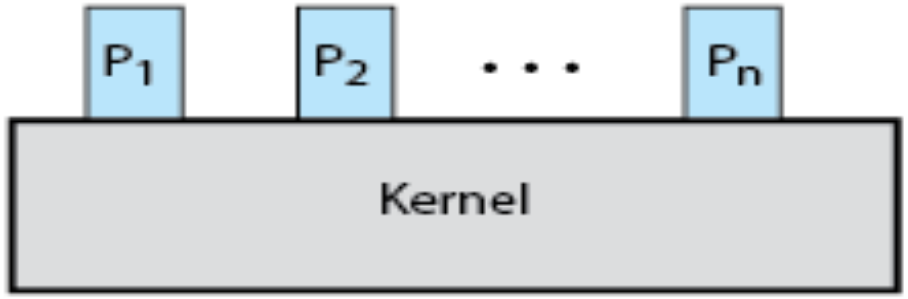


Change of Process State

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

Execution of the Operating System

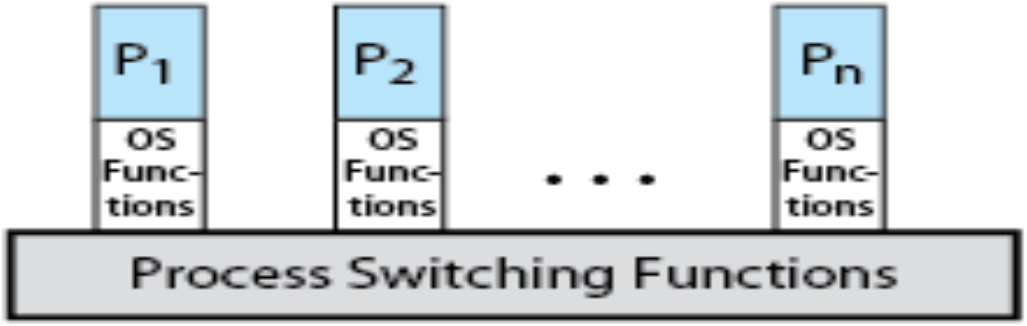
- Non-process Kernel
  - Execute kernel outside of any process
  - Operating system code is executed as a separate entity that operates in privileged mode



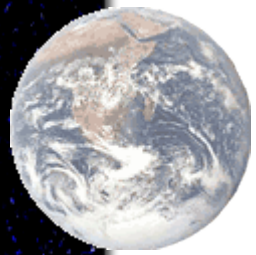
(a) Separate kernel

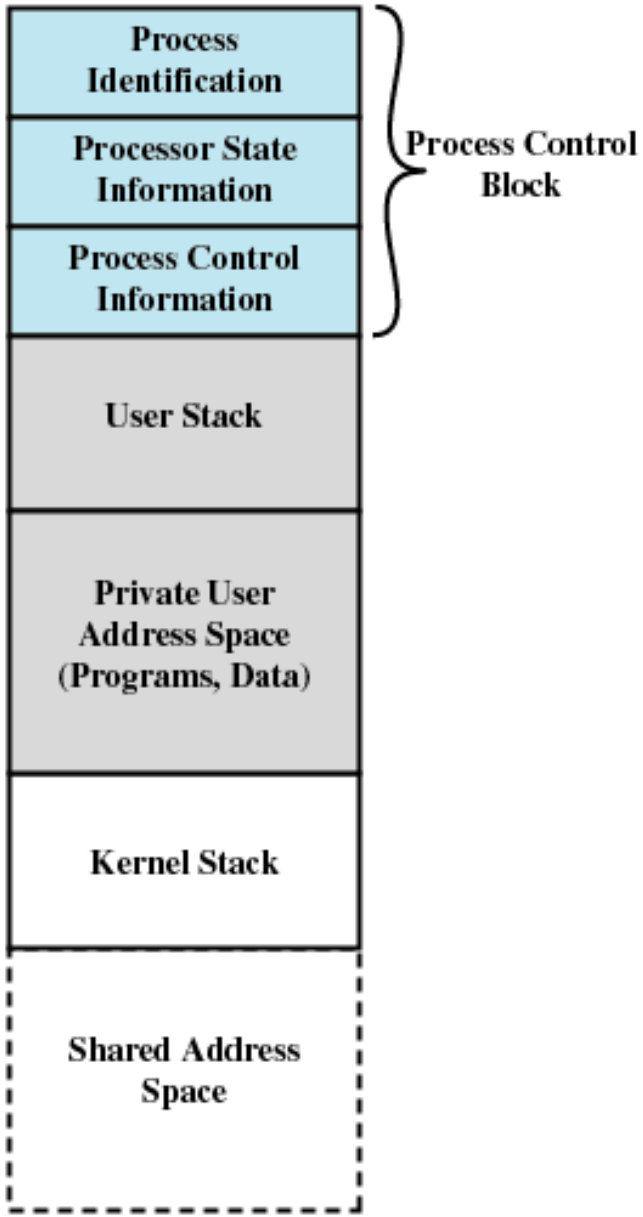
Execution of the Operating System

- Execution Within User Processes
  - Operating system software within context of a user process
  - Process executes in privileged mode when executing operating system code



(b) OS functions execute within user processes




Process Identification
Processor State Information
Process Control Information
Process Control Block
User Stack
Private User Address Space (Programs, Data)
Kernel Stack
Shared Address Space

Execution of the OS

- Process-Based Operating System
  - Implement operating system as a collection of system processes
  - Useful in multi-processor or multi-computer environment

(c) OS functions execute as separate processes