

UNIX and “UNIX -like” Operating Systems

Operating Systems In-Depth

Objectives of an operating system

- An operating system is a program that manages the computer's hardware.
 - It acts as an intermediate between users of a computer and the computer hardware.
 - It controls and coordinates the use of the hardware among the various application programs for the various users.
-

The 3 main purposes of an operating system

- To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.
 - To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
 - As a control program, it serves two major functions:
 - (1) supervision of the execution of user programs to prevent errors and improper use of the computer, and
 - (2) management of the operation and control of I/O devices.
-

Services provided by an Operating System

- Program execution
 - I/O Operation
 - File-System manipulation
 - Communications
 - Error detection
-

The Components of a Computer System

- Application Program
 - System Program
 - Operating System
 - Computer Hardware
-

The Kernel

A more common definition is that the OS is the one program running at all times on the computer, usually called the kernel, with all else being application programs.

Purpose of system calls

System calls allow user-level processes to request services of the operating system.

System programs can be thought of as bundles of useful system calls. They provide basic functionality to users so that users do not need to write their own programs to solve common problems..

5 major categories of System Calls

1. Process Control
2. File-management
3. Device-management
4. Information maintenance
5. Communications



System Calls in UNIX

- System calls define the programmer interface to UNIX
 - The set of systems programs commonly available defines the user interface
 - The programmer and user interface define the context that the kernel must support
 - Roughly three categories of system calls in UNIX
 1. File manipulation (same system calls also support device manipulation)
 2. Process control
 3. Information manipulation
-

5 major activities of process management

1. The creation and deletion of both user and system processes
 2. The suspension and resumption of processes
 3. The provision of mechanisms for process synchronization
 4. The provision of mechanisms for process communication
 5. The provision of mechanisms for deadlock handling
-

3 major activities of memory management

1. Keep track of which parts of memory are currently being used and by whom.
2. Decide which processes are to be loaded into memory when memory space becomes available.
3. Allocate and deallocate memory space as needed.

3 major activities of secondary storage management

1. Free-space management
2. Storage allocation
3. Disk scheduling

Interrupts and Traps

- An interrupt is a hardware-generated signal that changes the flow within the system.
 - A trap is a software-generated interrupt.
 - An interrupt can be used to signal the completion of I/O so that the CPU doesn't have to spend cycles polling the device.
 - A trap can be used to catch arithmetic errors or to call system routines
-

Interrupts and System Calls

- System calls provide the interface between a process and the operating system.
- When a system call is executed, it is treated by the hardware as a software interrupt

Process

- A process is a program in execution.
 - It is an active entity and it includes the process stack, containing temporary data and the data section contains global variables
 - A process is more than a program code, which is sometimes known as the text section. It also includes the current activity, as represented by the value of the program counter and the processor's registers.
-

Process States

- The state of the process is defined in part by the current activity of that process. Each process may be in one of the following states:
 - New: The process is being created.
 - Running: Instruction is being executed
 - Waiting: The process is waiting for some event to occur.
 - Ready: The process is waiting to be assigned to a processor
 - Terminated: The process has finished execution
-

Process control block

- Each process is represented in the OS by a process control block.
- It contains many pieces of information associated with a specific process.

Scheduler

- A process migrates between the various scheduling queues throughout its lifetime.
- The OS must select processes from these queues in some fashion.
- This selection process is carried out by a scheduler.

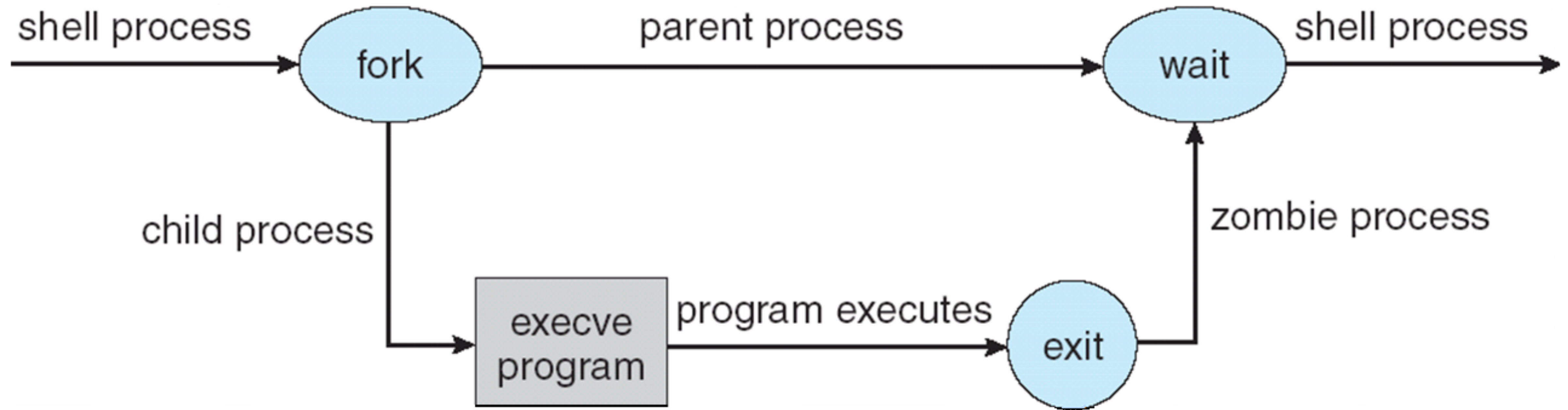
Job queues, Ready queues and Device queues

- As a process enters a system they are put into a job queue.
 - These queues consist of all jobs in the system.
 - The processes that are residing in main memory and are ready and waiting to execute are kept on a list called a ready queue.
 - The list of processes waiting for particular I/O devices is kept in the device queue.
-

Fork and exec system calls

- `fork()` is the name of the system call that the parent process uses to "divide" itself ("fork") into two identical processes.
 - After calling `fork()`, the Creating child process is an exact copy of the parent except for the return value.
 - When the child process calls `exec()`, all data in the original program is lost, and it is replaced with a running copy of the new program. This is known as overlaying.
 - Purpose:
 - Fork is a System call by which a new process is created.
 - Exec is also a System call, which is used after a fork by one of the two processes to replace the process memory space with a new program.
-

Illustration of Process Control Calls



Batch Systems

- Operators batched together jobs with similar needs and ran through the computer as a group.
- The operators would sort programs into batches with similar requirements and as system become available, it would run each batch.

Files

- A file is a named collection of related information that is recorded on secondary storage.
- A file contains either programs or data.
- A file has certain "structure" based on its type.

6 Basic File Operations

1. Creating a file
2. Writing a file
3. Reading a file
4. Repositioning within a file
5. Deleting a file
6. Truncating a file

Directory

- The device directory or simply known as the directory records information-such as name, location, size, and type for all files on that particular partition.
 - The directory can be viewed as a symbol table that translates file names into their directory entries.
-

Path Name

- A pathname is the path from the root through all subdirectories to a specified file.

