Lecture # 6

Topics: Process Control Block
Process States
Process transitions and State Diagram

Process is a program in execution. A program by itself is not a process. A program can be seen as a passive entity, such the content of a file, a process can be seen as an active entity.

Process Control Block
The PCB of a process provides to the OS all key information about the process. The PCB defines a process to the operating system. The access to the PCB of a process is done using a unique process ID.

Some of the information contained by the PCB:
Process State
Program Counter
CPU scheduling information (priority, pointers to scheduling queues)
Memory-management information (base and limit register, page tables, segment tables etc.)
Accounting Information
I/O status information (list of I/O devices allocated, list of open files etc.)
CPU registers (AC, PC, general-purpose registers) content

PROCESS CONTROL BLOCK INFORMATION

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

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. Typically, there are from 8 to 32 of these registers, although some RISC implementations have over 100.

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 operations (e.g., sign, zero, carry, equal, and overflow)
- *Status information:* includes interrupt enabled/disabled flags, execution mode.

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 Information

Scheduling and State Information

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

- *Process state:* defines the readiness of the process to be scheduled for execution (e.g., running, ready, waiting, halted)
- *Priority:* One or more fields may be used to describe the scheduling priority of the process. In some systems, several values are required (e.g., default, current, highest allowable).
- *Scheduling-related information:* this will depend on the scheduling algorithm used. Examples are the amount of time that the process has been waiting and the amount of time that the process executed the last time it was running.
- *Event:* identity of event the process is awaiting before it can be resumed.

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.

Inter-process 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.

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 fi1es. A history of utilization of the processor or other resources may also be included; this information may be need by the scheduler.

Interrupting a Process

- Asynchronous interrupt
- Synchronous interrupt
- System Call

Main processor actions (**mode switch**):
1. It saves the context of the processor.
2. It sets the PC to the starting address of an OS program. The OS checks the cause of the interrupt and calls the appropriate Interrupt Service (Handler) routine to service the interrupt. One of the main tasks of the Interrupt Handler is to protect the PCB of the process. Usually it is the only process that can modify information inside the PCB.
Meanwhile the processor switched from the user mode to the system mode.

If the interrupt results in a full Process Switch (**full context switch**)
1. save the context of the processor.
2. update the PCB of the process that is currently in the Running state.
3. move the PCB of this process to the appropriate queue.
4. select another process for execution.
5. update the PCB of the new process.
6. update memory management data structures.
7. restore the context of the processor to that which existed at the time the selected process was last switched out of the Running state.

Process Switch times are pure overhead and are highly dependent on hardware support.

Process States
New - a process that just has been created but has not been yet admitted to the pool of executable processes.
Ready swapped - the process is in secondary memory, but is available for execution as soon as it is loaded into main memory.
Ready (active) - the process is in main memory and available for execution.
Running - the process is currently being executed.
Blocked - the process is in main memory and awaiting an event. The process doesn't have all the resources it needs.
Blocked swapped - the process is in secondary memory and it may wait for an event.
Terminated - the process has finished execution.

State Transitions & State Diagram (covered in class)