Table 3.5 Typical Elements of a 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

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.

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

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:

- **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.

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.

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.

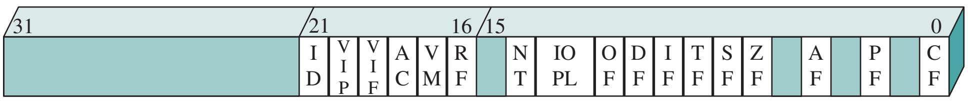
With respect to **process identification**, in virtually all operating systems, each process is assigned a unique numeric identifier, which may simply be an index into the primary process table (Figure 3.11); otherwise there must be a mapping that allows the OS to locate the appropriate tables based on the process identifier. This identifier is useful in several ways. Many of the other tables controlled by the OS may use process identifiers to cross-reference process tables. For example, the memory tables may be organized so as to provide a map of main memory with an indication of which process is assigned to each region. Similar references will appear in I/O and file tables. When processes communicate with one another, the process identifier informs the OS of the destination of a particular communication. When processes are allowed to create other processes, identifiers indicate the parent and descendents of each process.

In addition to these process identifiers, a process may be assigned a user identifier that indicates the user responsible for the job.

Processor state information consists of the contents of processor registers. While a process is running, of course, the information is in the registers. When a process is interrupted, all of this register information must be saved so that it can be restored when the process resumes execution. The nature and number of registers involved depend on the design of the processor. Typically, the register set will include user-visible registers, control and status registers, and stack pointers. These are described in Chapter 1.

Of particular note, all processor designs include a register or set of registers, often known as the program status word (PSW), that contains status information. The PSW typically contain condition codes plus other status information. A good example of a processor status word is that on Intel x86 processors, referred to as the EFLAGS register (shown in Figure 3.12 and Table 3.6). This structure is used by any OS (including UNIX and Windows) running on an x86 processor.

The third major category of information in the process control block can be called, for want of a better name, **process control information**. This is the additional information needed by the OS to control and coordinate the various active processes. The last part of Table 3.5 indicates the scope of this information. As



ID	= Identification flag	DF	= Direction flag
VIP	= Virtual interrupt pending	IF	= Interrupt enable flag
VIF	= Virtual interrupt flag	TF	= Trap flag
AC	= Alignment check	SF	= Sign flag
VM	= Virtual 8086 mode	ZF	= Zero flag
RF	= Resume flag	AF	= Auxiliary carry flag
NT	= Nested task flag	PF	= Parity flag
IOPL	= I/O privilege level	CF	= Carry flag
OF	= Overflow flag		

Figure 3.12 x86 EFLAGS Register

Table 3.6 Pentium EFLAGS Register Bits

Control Bits	
AC (Alignment check)	Set if a word or doubleword is addressed on a nonword or non-doubleword boundary.
ID (Identification flag)	If this bit can be set and cleared, this processor supports the CPUID instruction. This instruction provides information about the vendor, family, and model.
RF (Resume flag)	Allows the programmer to disable debug exceptions so that the instruction can be restarted after a debug exception without immediately causing another debug exception.
IOPL (I/O privilege level)	When set, causes the processor to generate an exception on all accesses to I/O devices during protected mode operation.
DF (Direction flag)	Determines whether string processing instructions increment or decrement the 16-bit half-registers SI and DI (for 16-bit operations) or the 32-bit registers ESI and EDI (for 32-bit operations).
IF (Interrupt enable flag)	When set, the processor will recognize external interrupts.
TF (Trap flag)	When set, causes an interrupt after the execution of each instruction. This is used for debugging.
Operating Mode Bits	
NT (Nested task flag)	Indicates that the current task is nested within another task in protected mode operation.
VM (Virtual 8086 mode)	Allows the programmer to enable or disable virtual 8086 mode, which determines whether the processor runs as an 8086 machine.
VIP (Virtual interrupt pending)	Used in virtual 8086 mode to indicate that one or more interrupts are awaiting service.
VIF (Virtual interrupt flag)	Used in virtual 8086 mode instead of IF.
Condition Codes	
AF (Auxiliary carry flag)	Represents carrying or borrowing between half-bytes of an 8-bit arithmetic or logic operation using the AL register.
CF (Carry flag)	Indicates carrying out or borrowing into the leftmost bit position following an arithmetic operation. Also modified by some of the shift and rotate operations.
OF (Overflow flag)	Indicates an arithmetic overflow after an addition or subtraction.
PF (Parity flag)	Parity of the result of an arithmetic or logic operation. 1 indicates even parity; 0 indicates odd parity.
SF (Sign flag)	Indicates the sign of the result of an arithmetic or logic operation.
ZF (Zero flag)	Indicates that the result of an arithmetic or logic operation is 0.