Inter-Process Communication and Synchronization of Processes, Threads and Tasks:

Lesson-1: PROCESS

Process Concepts

Process

- A process consists of executable program (codes), *state* of which is controlled by OS,
- The *state* during running of a process—represented by process-status (running, blocked, or finished), process-structure—its data, objects and resources, and process control block (PCB).

Process….

- Runs when it is scheduled to run by the OS (kernel)
- OS gives the control of the CPU on a process's request (system call).
- Runs by executing the instructions and the continuous changes of its state takes place as the program counter (PC) changes

Process …

- Process is that executing unit of computation, which is controlled by some process (of the OS) for a scheduling mechanism that lets it execute on the CPU and by some process at OS for a resource-management mechanism that lets it use the system-memory and other system-resources such as network, file, display or printer

Application program can be said to consist of number of processes

Example – Mobile Phone Device embedded software

- Software highly complex.
- Number of functions, ISRs, processes threads, multiple physical and virtual device drivers, and several program objects that must be concurrently processed on a single processor.

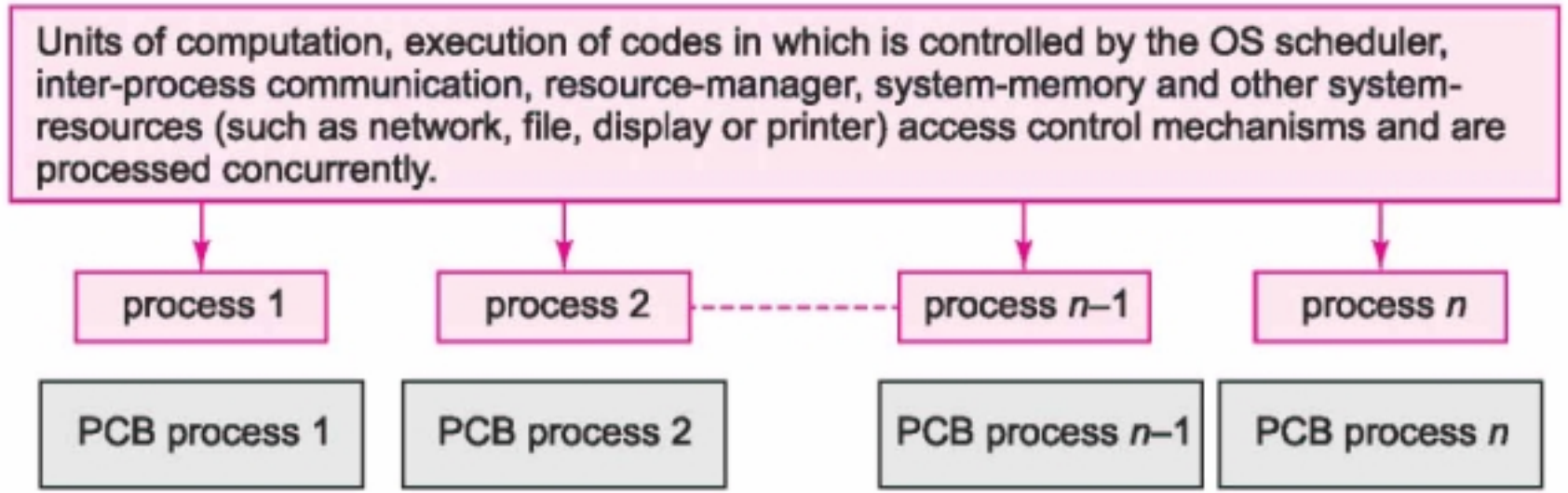
Exemplary processes at the phone device

- Voice encoding and convoluting process— the device captures the spoken words through a speaker and generates the digital signals after analog to digital conversion, the digits are encoded and convoluted using a CODEC,
- Modulating process,
- Display process,

Exemplary processes at the phone device

- GUIs (graphic user interfaces), and
- Key input process — for provisioning of the user interrupts

Process


Units of computation, execution of codes in which is controlled by the OS scheduler, inter-process communication, resource-manager, system-memory and other system-resources (such as network, file, display or printer) access control mechanisms and are processed concurrently.
process 1
process 2
process n–1
process n
PCB process 1
PCB process 2
PCB process n–1
PCB process n

Process Control Block

Process Control Block

- A data structure having the information using which the OS controls the process state.
- Stores in protected memory area of the kernel.
- Consists of the information about the process state

Information about the process state at Process Control Block…

- Process ID,
- process priority,
- parent process (if any),
- child process (if any), and
- address to the next process PCB which will run,

Information about the process state at Process Control Block…

- allocated program memory address blocks in physical memory and in secondary (virtual) memory for the process-codes,
- allocated process-specific data address-blocks

Information about the process state at Process Control Block…

- allocated process-heap (data generated during the program run) addresses,
- allocated process-stack addresses for the functions called during running of the process,

Information about the process state at Process Control Block…

- allocated addresses of CPU register-save area as a process context represents by CPU registers, which include the program counter and stack pointer
- allocated addresses of CPU register-save area as a process context

[Register-contents (define process context) include the program counter and stack pointer contents]

Information about the process state at Process Control Block…

- process-state signal mask [when mask is set to 0 (active) the process is inhibited from running and when reset to 1, the process is allowed to run],
- Signals (messages) dispatch table [process IPC functions],

Information about the process state at Process Control Block…

- OS allocated resources' descriptors (for example, file descriptors for open files, device descriptors for open (accessible) devices, device-buffer addresses and status, socket-descriptor for open socket), and
- Security restrictions and permissions.

Context

- Context loads into the CPU registers from memory when process starts running, and the registers save at the addresses of register-save area on the context switch to another process

Context

- The present CPU registers, which include program counter and stack pointer are called context
- When context saves on the PCB pointed process-stack and register-save area addresses, then the running process stops.
- Other process context now loads and that process runs— This means that the context has switched

Summary

We learnt

- Application program can be said to consist of number of processes
- Process defined as that executing unit of of computation that processes on a CPU and state of which is under the control of kernel of an operating system.

We learnt

- Process state at an instance defines by process-status (running, blocked, or finished), process-structure—its data, objects and resources and process control block.

We learnt

- OS lets a process execute on the CPU—some process at OS for a resource-management mechanism lets it use the system-memory and other system-resources such as network, file, display or printer

We learnt

- PCB– a data structure having the information using which the OS controls the process state
- PCB consists of the information about the process state
- PCB stores in protected memory area of the kernel

End of Lesson 1 of Chapter 7