



# **SNS COLLEGE OF TECHNOLOGY**

**Coimbatore-35**  
**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



## **DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

### **19ITT204 – MICROCONTROLLER & EMBEDDED SYSTEMS**

III YEAR - V SEM

UNIT 4 – Processes and Operating Systems

Topic-Task, Process and Thread



# Process



- 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)
- Runs when it is scheduled to run by the OS(kernel)
- OS gives the control of the CPU and resources on system call on a Process's request.



# Process



- Runs by executing the instructions and the continuous changes of its state takes place as the program counter (PC) changes.
- A process resource – management mechanism that lets it use the system- memory and other system- resources such as network, file, display or printer.



# Process Control Block



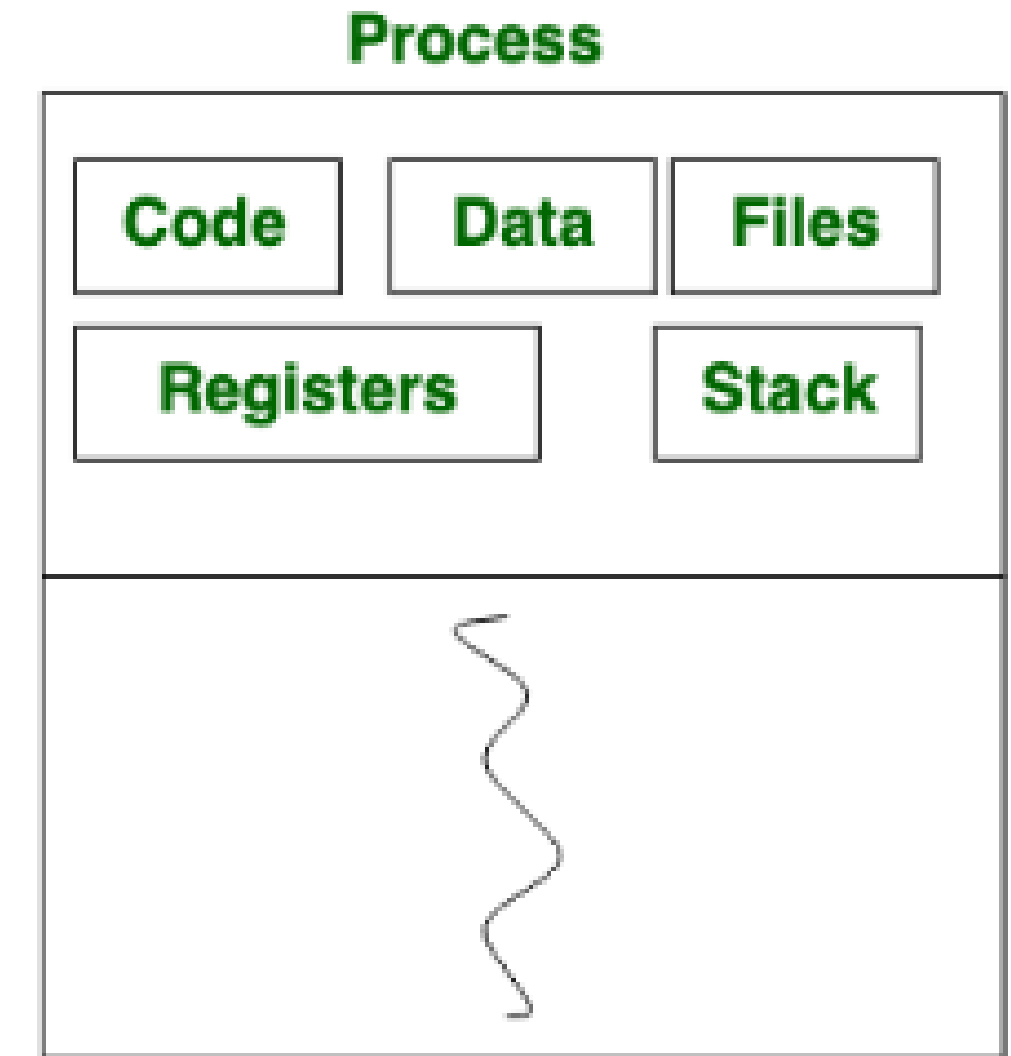
- Consists of the information about the process state
- A data structure having the information of the process (not accessible to other process) other than the kernel
- OS controls the state of the process using that information
- The information stores in protected memory area.



# Thread



- Thread runs like a process and consists of executable program (codes), state of which is controlled by OS.
- A thread has three states: Running, Ready, and Blocked.
- The [thread](#) takes less time to terminate as compared to the process but unlike the process, threads do not isolate.



**Thread**

*Process vs Thread*



# Thread



*A thread* is a process or sub-process within a process that has its own program counter, its own stack pointer, and stack, its own priority-parameter for its scheduling by a thread-scheduler, and its own variables that load into the processor registers on context switching and is processed concurrently along with other threads.



# Thread



- An application program can be said to consist of number of threads.
- A process can be said to consist of number of threads in Unix, Linux and other OSes
- Each thread has independent parameters- ID, priority, program counter, stack pointer, CPU registers and its present status.
- When the function in a thread in OS is called, the calling function state is placed on the stack top.
- When there is return the calling function takes the state information from the stack top.



# Thread



- An application program can be said to consist of number of threads.
- A process can be said to consist of number of threads in Unix, Linux and other OSes
- Each thread has independent parameters- ID, priority, program counter, stack pointer, CPU registers and its present status.
- When the function in a thread in OS is called, the calling function state is placed on the stack top.
- When there is return the calling function takes the state information from the stack top.





# Difference between process and thread



S.NO	Process	Thread
1.	Process means any program is in execution.	Thread means a segment of a process.
2.	The process takes more time to terminate.	The thread takes less time to terminate.
3.	It takes more time for creation.	It takes less time for creation.
4.	It also takes more time for context switching.	It takes less time for context switching.
5.	The process is less efficient in terms of communication.	Thread is more efficient in terms of communication.
6.	Multiprogramming holds the concepts of multi-process.	We don't need multi programs in action for multiple threads because a single process consists of multiple threads.
7.	The process is isolated.	Threads share memory.
8.	The process is called the heavyweight process.	A Thread is lightweight as each thread in a process shares code, data, and resources.



# Task



- A task is a process and the OS does the multitasking
- Task is a kernel controlled entity while thread is a process controlled entity
- A task does not directly call another task to run
- Multitasking also needs a task-scheduler
- There may or may not be task groups and task libraries in a given OS



# References

<https://www.javatpoint.com/embedded-operating-system>

<https://www.tutorialspoint.com/what-is-an-embedded-operating-system>

<https://blog.felgo.com/embedded/embedded-operating-systems>

Rajkamal, Embedded system, Tata McGraw-Hill Publishers ,2<sup>nd</sup> edition,2008

*Thank You*