Program counter ----- address of next instruction to execute

CPU registers ----- contents of all CPU registers

Process scheduler -- selects among ready processes for next execution on CPU core

User-level thread

A user thread is not created by the OS kernel. —— T

A user thread is scheduled by the OS kernel. —— F

A user thread cannot make a system call. —— F

Context of a user thread is stored in the Process Control Block (PCB). —— F

multi-thread process

All the threads within the process share one single data segment. —— T

All the threads within the process share one single heap segment. —— T

Each thread has its own thread control block. —— T

Each thread has its own stack space. —— T

processes and program files?

A process can execute at most one program file during its lifetime.

A process can execute multiple program files during its lifetime. ——- T

A program file can only be executed in at most one process at a time.

A program file can be executed in multiple processes simultaneously. —— T

An interrupt service routine is executed on the processor. --- True

Concurrent execution of threads is possible with many-to-one mapping –-- T

Every thread has a unique stack. --- T

Amdahl's Law describes potential performance gains for applications with both serial and parallel components. --- T

FCFS achieves similar average wait times regardless of process arrival times. --- T

A system call is triggered by hardware. --- F

System calls are used to interface with the OS because they are faster than function calls. --- F

The process id is changed after a process becomes orphaned. --- F

The exec( ) system call creates a new process. --- F

When fork( ) is invoked, it is passed a set of flags that determine how much sharing is to take place between the parent and child tasks. --- F

The operating system switches kernel threads by switching thread IDs. --- F

A Shell converts all user inputs into system calls. —— F

Processes all share the system stack when they are in user mode. —— F

In Symmetric Multi-Processor (SMP) Scheduling, soft affinity does not allow a process to migrate between processors during its execution. —— F

It is more expensive (in terms of resources used) for new thread creation than new process creation. —— F

can cause a non-preemptive scheduler to swap in a process to run on the CPU

A process changes from RUNNING to WAITING.

A process changes from RUNNING to TERMINATED.

Which of the following is (are) true about the execution of “printf(``Hello!’’);” in a Linux system?

A function of the C library is called.

A system call is made to the kernel.

The execution mode switches from “user” to “kernel”.

The execution mode switches from “kernel” to “user”.