

Introducción a Concurrencia

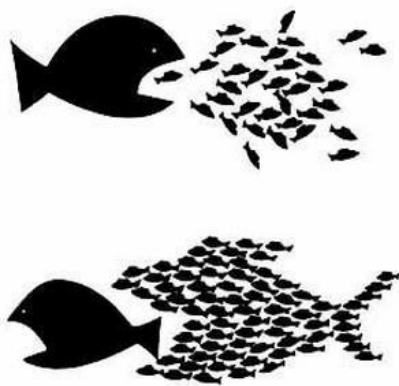
Conceptos básicos

Oscar A. Esquivel-Flores

LCD-UNAM

23 de agosto de 2019

¿Divide y vencerás?



Introducción I

- El avance del hardware ha proporcionado arquitecturas con más recursos utilizados en la mayoría de los dispositivos.
- Aumentan los procesadores en las computadoras personales
- Se incrementan los clusters de computadoras
- Los servicios de cómputo en la nube son más populares y dirigidos a científicos y tecnólogos

Introducción I

- Actualmente no es suficiente solo utilizar estos recursos, es importante desarrollar programas que utilicen de mejor manera procesadores, memoria, velocidad de procesamiento, etc.
- Sin embargo, se tiene aún poca experiencia y mucho menos aplicaciones del cómputo eficiente.
- El uso optimizado de los recursos computacionales puede ofrecer grandes beneficios y mejoras en muchos campos del cómputo científico.

Introducción I

- El aumento en la disponibilidad de los procesadores tiene una consecuencia importante en los desarrolladores de software.
- Agregar más recursos no conducen de inmediato a mejorar el “rendimiento” en la mayoría de los programas “seriales”.
- ¿Es posible crear programas que tomen ventaja de los recursos disponibles?
- Incrementar el **rendimiento** (*performance*)

Serial, Concurrencia, Paralelismo, Dtribuido

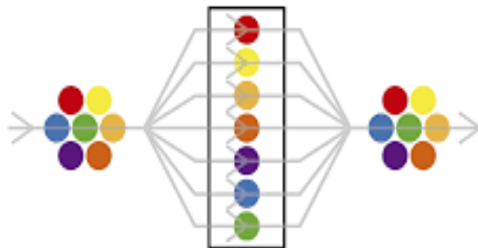
- **Cómputo secuencial:** Un programa en el cual múltiples tareas se ejecutan una detrás de otra.
- **Cómputo concurrente:** Un programa en el cual múltiples tareas pueden estar en proceso en cualquier instante.
- **Cómputo paralelo:** Un programa en el cual múltiples tareas cooperan de manera conjunta para resolver un problema.
- **Cómputo distribuido:** Un programa el cual puede necesitar cooperar con otros programas para resolver un problema.

Serial, Concurrencia, Paralelismo, Distribuido

Procesamiento en serie



Procesamiento en paralelo



Procesos

- **Programa:** Es una colección de instrucciones y de datos que se encuentran almacenados en un archivo.
- Este archivo puede ser ejecutable dependiendo de los permisos que tenga.
- Cuando el archivo es leído del disco por el núcleo del sistema operativo y cargado en memoria se convierte en un proceso

Procesos

- **Procesos:** Cuando un usuario ejecuta un programa, el sistema operativo crea un proceso, una instancia de un programa de cómputo se está ejecutando:
 - Programa en lenguaje máquina
 - Bloque de memoria
 - Descriptores de recursos
 - Información de seguridad
 - Información del estado de los procesos

Multitareas

- **Multitasking:** El sistema ofrece facilidades para ejecutar múltiples programas aparentemente al mismo tiempo (simultáneamente), aún en un sistema mono-core.
 - Esto es posible pues cada proceso se ejecuta en un pequeño intervalo de tiempo (*ms*).
 - Distintos programas son ejecutados en una ventana de tiempo (*time slicing*).
- El S.O. puede cambiar la ejecución de los procesos varias veces, administrar recursos y pausar procesos (blocking).

Hilos

- **Hilo** (*thread*): Son ejecuciones de “peso ligero” que permiten facilitar al S.O. el intercambio entre ellas.
 - Los hilos están contenidos dentro de los procesos, comparten memoria y los mismos dispositivos de I/O.
 - Dos hilos pueden compartir la mayoría de los recursos del proceso. Cada hilo requiere su propio contador de programa y su propia llamada al stack, así pueden ejecutarse uno independientemente del otro.

Threading

- **Threading:** Provee a los programadores un mecanismo para dividir los programas en tareas más o menos independientes con la propiedad que mientras un hilo es pausado otro puede ejecutarse.

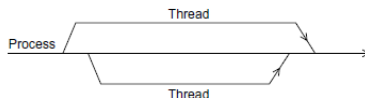


FIGURE 2.2

A process and two threads

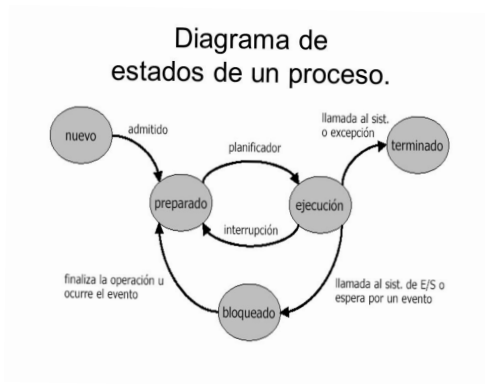
Procesos y estados

- Un proceso se compone de tres bloques o segmentos:
 - **Segmento de texto:** Contiene instrucciones que comprende la UCP. Es una copia del bloque de texto del programa.
 - **Segmento de datos:** Contiene los datos que deben ser inicializados al arrancar el proceso, aquí estarán las variables globales y estáticas.
 - **Segmento de pila:** Lo crea el núcleo al arrancar el proceso y su tamaño es gestionado por el núcleo. Compuesto por varios *marcos de pila* introducidos por la llamada a una función.

Procesos y estados

- El tiempo de vida de un proceso se puede dividir en un conjunto de estados:
 - El proceso se está ejecutando.
 - El proceso no se está ejecutando, está listo para ejecutarse.
 - El proceso está dormindo. Se encuentra en espera que se complete una operación de entrada salida.

Procesos y estados



Llamadas al sistema

- UNIX es un núcleo/kernel de un sistema operativo de tiempo compartido.
 - Controla recursos hardware
 - Controla I/O
 - Proporciona sistema de archivos
- Un programa necesita un medio para tener interacción con el hardware.
- Las **llamadas al sistema** (*system calls*) es una interfaz entre el sistema operativo y el programador.
- Cualquier programa ejecutado bajo UNIX-LINUX que necesite hacer uso de cualquiera de los recursos deberá hacer una *llamada al sistema*

Procesos

- Un programa es una colección de instrucciones y datos almacenados en un archivo.
- Cuando un programa es leído por el núcleo y se carga en memoria para ejecutarse se convierte en **proceso**
- El tiempo de vida de un proceso se puede dividir en un conjunto de estados. En un modelo de tres estados se tiene:
 - Ejecutando
 - Listo para ejecución
 - Bloqueado

Creación de un proceso

- La creación de un proceso significa:
 - Dar nombre al proceso
 - Insertar un proceso en la lista del sistema de procesos conocidos.
 - Determinar la prioridad inicial del proceso
 - Crear el bloque de control del proceso
 - Asignar los recursos iniciales del proceso
- Un proceso puede crear un nuevo proceso
- El proceso creador se denomina **proceso padre**
- El proceso creado se denomina **proceso hijo**
- Se obtiene así una estructura jerárquica de procesos

Hilos

- Los procesos tienen dificultad para relacionarse con otros procesos.
- Los hilos no sustituyen los procesos, los enriquecen
- Un proceso es una entidad compuesta por recursos e hilos.
 - Los recursos se refieren a los datos del proceso.
 - Los **hilos** son objetos dinámicos que ejecutan instrucciones y comparten recursos del proceso.
- Un *proceso ligero* es un hilo del usuario sostenido por el núcleo
- Un proceso se puede componer de varios procesos ligeros, apoyados cada uno en un hilo del núcleo.

POSIX

- **Portable Operating System Interface**
- Pthreads

POSIX

- El comando *ps* muestra la información sobre procesos

Campo	Significado
UID	ID del usuario propietario del proceso
PID	ID del proceso
PPID	ID del padre del proceso
C	Utilización del preprocesador de C para la administración de procesos
STIME	Hora de comienzo
TTY	terminal de control
TIME	Tiempo acumulado de CPU
CMD	Nombre del comando

Python

- multiprocessing
- threading