

Programación multiproceso

Programación de servicios y procesos
(octubre 2019)

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Definiciones

- Ejecutable: programa reconocido por el SO y puede hacerlo correr
 - Windows: .exe
 - Android: .apk
 - Scripts: .sh, .bat
 - Java: .jar, .war, .ear
 - ...

Definiciones

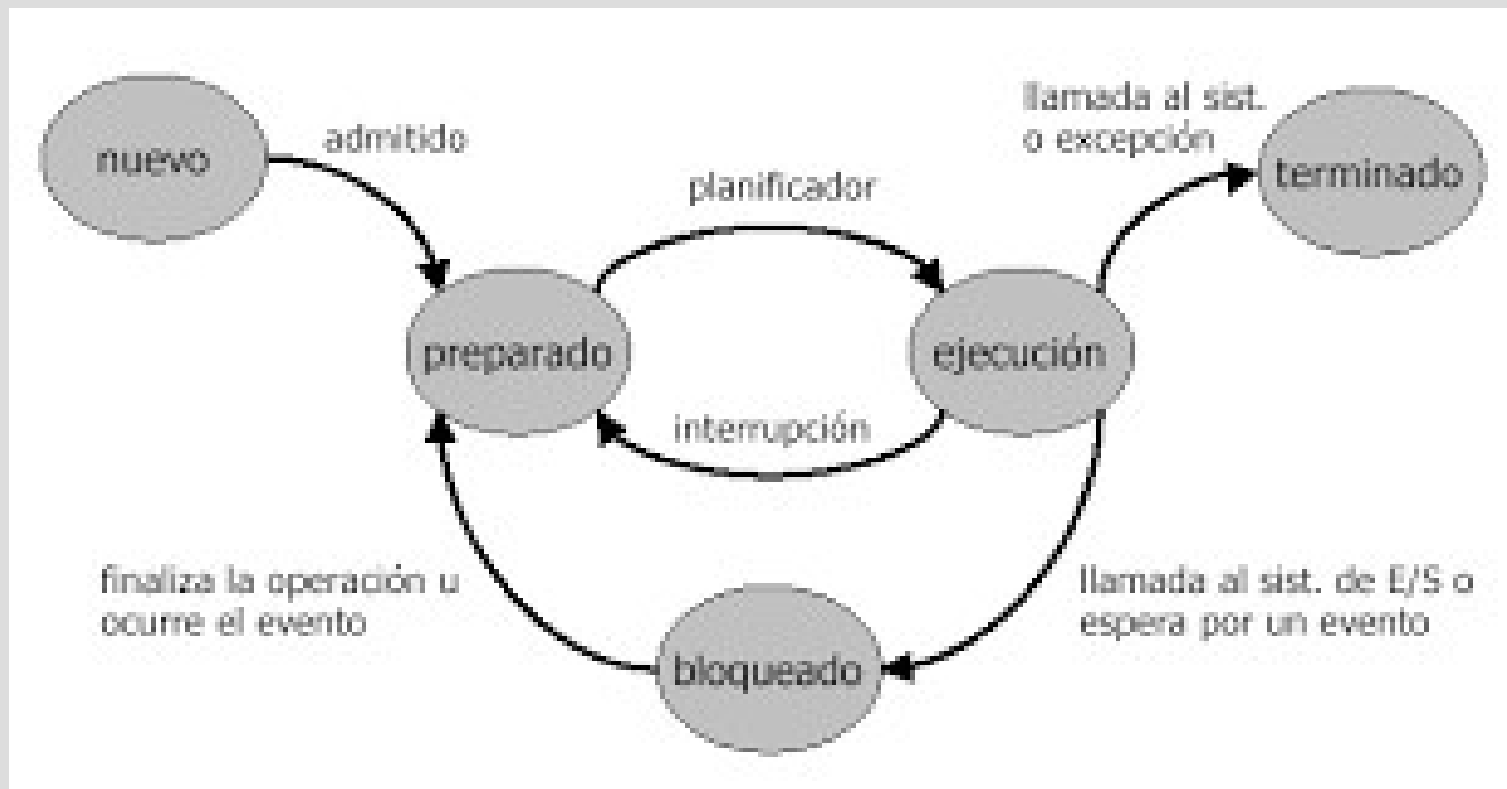
- Proceso: programa en ejecución
 - Definición formal: ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados
- Son gestionados por el sistema operativo
- Cada proceso tiene un PID que los identifica y es asignado cuando se crea el proceso

Programación multiproceso

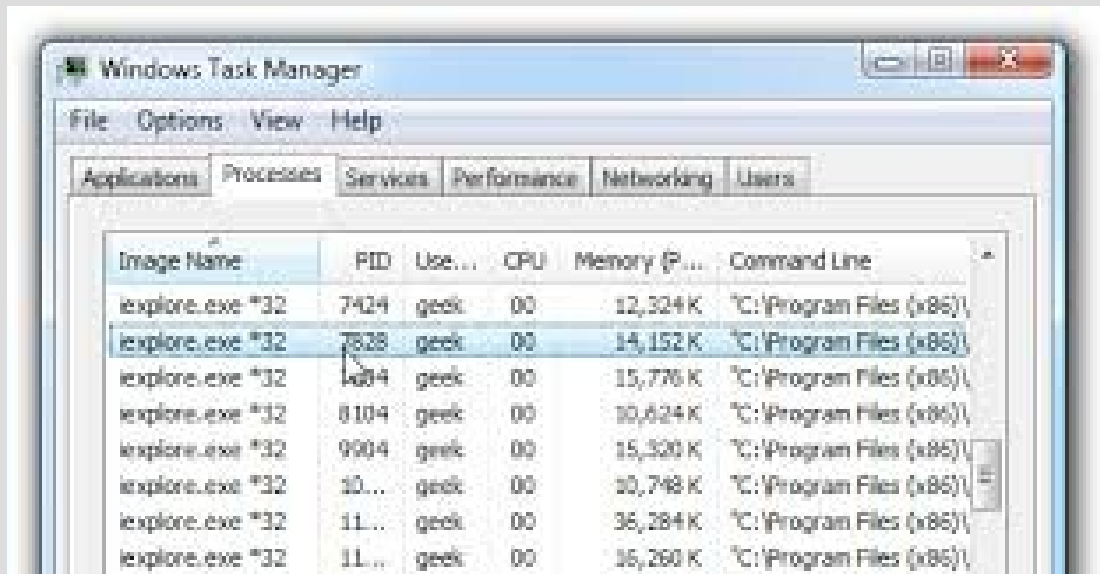
- Definiciones
- **Procesos**
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Procesos

- Estados de un proceso:



Procesos



Processes: 314 total, 2 running, 312 sleeping, 1603 threads
 Load Avg: 2.10, 2.28, 2.47 CPU usage: 3.85% user, 6.74% sys, 89.39% idle SharedLibs: 179M resident, 53M data, 20M linkedit.
 MemRegions: 90515 total, 2734M resident, 53M private, 1188M shared. PhysMem: 8036M used (1752M wired), 155M unused.
 VM: 1460G vsize, 1371M framework vsize, 31347954(0) swapins, 32162728(0) swapouts. Networks: packets: 8287334/9081M in, 5655257/605M out.
 Disks: 5517499/165G read, 2468494/143G written.

11

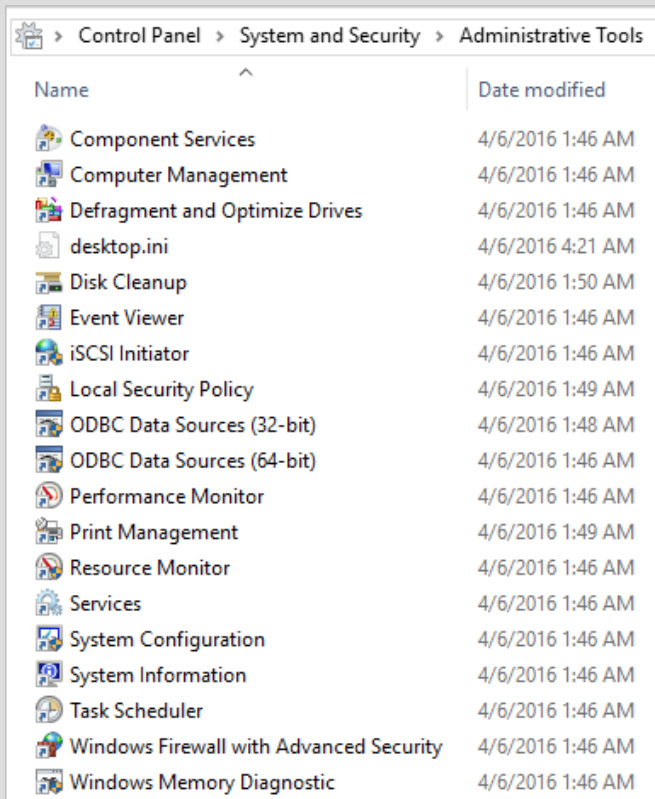
PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	%CPU_ME	%CPU_OTHR	UID	FAULTS	COW
24052	screencaptur	0.1	00:00.14	5	3	152	3992K	0B	0B	24052	1	sleeping	*0[260+]	0.00000	0.02192	502	5658	311
24051	screencaptur	0.8	00:00.66	2	1	54	2908K	620K	0B	298	298	sleeping	*0[1]	0.02192	0.00000	502	28572+	193
24047	top	3.2	00:01.27	1/1	0	27	3368K	0B	0B	24047	8414	running	*0[1]	0.00000	0.00000	0	8859+	102
23714	ScopedBookma	0.0	00:00.05	2	2	45	2004K	0B	1120K	23714	1	sleeping	0[0]	0.00000	0.00000	502	2197	146
23707	mdworker	0.0	00:00.09	3	1	59	3500K	0B	2388K	23707	1	sleeping	*0[1]	0.00000	0.00000	502	4222	195
23605	mdworker_sha	0.0	00:01.12	4	1	61	9328K	0B	3444K	23605	1	sleeping	*0[1]	0.00000	0.00000	502	44283	189
23604	mdworker_sha	0.0	00:00.94	4	1	58	8520K	0B	2416K	23604	1	sleeping	*0[1]	0.00000	0.00000	502	43110	187
23601	mdworker_sha	0.0	00:01.00	4	1	61	8736K	0B	3660K	23601	1	sleeping	*0[1]	0.00000	0.00000	502	38028	189
23553	plugin-conta	0.1	00:38.29	27	1	190	234M	0B	42M	956	956	sleeping	*0[62]	0.00000	0.00000	502	920878	7181
23513	netbiosd	0.0	00:00.06	2	2	26	2340K	0B	1996K	23513	1	sleeping	*0[1]	0.00000	0.00000	222	3103	159
23357	MTLCompilerS	0.0	00:00.12	2	2	22	7184K	0B	7172K	23357	1	sleeping	0[5]	0.00000	0.00000	502	5298	455
23340	CoreServices	0.0	00:00.31	3	1	153	3964K	0B	2208K	23340	1	sleeping	*0[1]	0.00000	0.00000	502	5568	256
23189	VTDecoderXPC	0.0	00:01.11	2	1	51	5224K	0B	4940K	23189	1	sleeping	0[11]	0.00000	0.00000	502	7891	208
23179	VTDecoderXPC	0.0	00:18.95	2	1	51	4664K	0B	4356K	23179	1	sleeping	0[5]	0.00000	0.00000	502	43930	208
23093	MTLCompilerS	0.0	00:00.06	2	2	24	6000K	0B	5988K	23093	1	sleeping	0[2]	0.00000	0.00000	502	4586	455
22834	IMDPersisten	0.0	00:00.14	2	1	66	2352K	0B	2080K	22834	1	sleeping	0[12]	0.00000	0.00000	502	5593	188

Programación multiproceso

- Definiciones
- Procesos
- **Servicios**
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Servicios

- Servicio: programas o aplicaciones cargadas por el propio sistema operativo



The screenshot shows the 'Administrative Tools' window in Windows. The title bar reads 'Control Panel > System and Security > Administrative Tools'. The window contains a list of system utilities with columns for 'Name' and 'Date modified'. The 'Name' column is sorted in ascending order. The 'Date modified' column shows the last modification date and time for each utility.

Name	Date modified
Component Services	4/6/2016 1:46 AM
Computer Management	4/6/2016 1:46 AM
Defragment and Optimize Drives	4/6/2016 1:46 AM
desktop.ini	4/6/2016 4:21 AM
Disk Cleanup	4/6/2016 1:50 AM
Event Viewer	4/6/2016 1:46 AM
iSCSI Initiator	4/6/2016 1:46 AM
Local Security Policy	4/6/2016 1:49 AM
ODBC Data Sources (32-bit)	4/6/2016 1:48 AM
ODBC Data Sources (64-bit)	4/6/2016 1:46 AM
Performance Monitor	4/6/2016 1:46 AM
Print Management	4/6/2016 1:49 AM
Resource Monitor	4/6/2016 1:46 AM
Services	4/6/2016 1:46 AM
System Configuration	4/6/2016 1:46 AM
System Information	4/6/2016 1:46 AM
Task Scheduler	4/6/2016 1:46 AM
Windows Firewall with Advanced Security	4/6/2016 1:46 AM
Windows Memory Diagnostic	4/6/2016 1:46 AM

Corre en background,

Sin interfaz de usuario

Proveen de características del core del SO (Web serving, event logging, file serving, printing, cryptography, error reporting).

Windows: panel de control → herramientas administrativas → servicios

Unix: se llaman demonios. Comando `systemctl/service`

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- **Hilos**
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Hilos (threads)

- Hilo – thread – hebra
- Llamados sub-proceso o proceso ligero
 - secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo.
- Comparten recursos (memoria)
- Un proceso puede lanzar varios hilos
- Propio de cada hilo: contador de programa, la pila de ejecución y el estado de la CPU (incluyendo el valor de los registros).

Hilos (threads)

- Thread
 - Prioridad: Cada hilo tiene una prioridad.
 - Con prioridad alta se ejecutan preferentemente a los de baja prioridad.
 - $MAX < 10$
 - $MIN > 1$

Hilos (threads)

- Java
 - Interfaz Runnable
 - extends Thread

```
class PrimeRun implements Runnable {
    long minPrime;
    PrimeRun(long minPrime) {
        this.minPrime = minPrime;
    }

    public void run() {
        // compute primes larger than minPrime
        . . .
    }
}
```

```
class PrimeThread extends Thread {
    long minPrime;
    PrimeThread(long minPrime) {
        this.minPrime = minPrime;
    }

    public void run() {
        // compute primes larger than minPrime
        . . .
    }
}
```

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- **Programación concurrente**
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Programación concurrente

- Cuando dos o más tareas se realizan al mismo tiempo
- Objetivo: tener algún proceso en ejecución en todo momento (maximizar el aprovechamiento de la CPU).
- Puede ejecutarse
 - Sobre un único procesador o core
 - Multiprocesador o multicore
 - Computación distribuida

Programación concurrente

- Puede ejecutarse
 - Sobre un único procesador o core
 - Multiprocesador o multicore
 - Computación distribuida

Programación concurrente

- Programación paralela:
 - en un único ordenador
- Programación distribuida:
 - dentro de una red de computadores (clúster, grid, cloud)

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- **Programación paralela**
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Programación paralela

- Se necesitan varios procesadores o núcleos
- Se ejecutan varias instrucciones a la vez en diferentes procesadores
- Mayor complejidad
 - Comunicación entre procesos
 - Sincronización

Concurrencia vs paralelismo

Concurrencia

Paralelismo

CPU

CPU

CPU



CPU



Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- **Programación distribuida**
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Programación distribuida

- Desarrollo de sistemas distribuidos:
 - abiertos,
 - escalables,
 - transparentes y
 - tolerantes a fallos.
- Evolución natural

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- **Creación de procesos**
- Comunicación entre procesos
- Gestión de procesos
- Sincronización

Creación de procesos

- Creación de procesos
 - Al arrancar el sistema
 - Por otro proceso
 - Por acción del usuario
 - Por un trabajo por lotes
- Creador: proceso padre
- Creado: proceso hijo

Creación de procesos

- Ciclo de vida: el padre terminará cuando todos sus hijos hayan terminado
- Sistemas UNIX:
 - Función `fork()`: se crea un proceso hijo que es una copia del padre. Ambos procesos continúan ejecutándose desde el punto en el que se hizo la llamada a `fork()`.
 - Función `getpid()`: retorna el PID de un proceso

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- **Comunicación entre procesos**
- Gestión de procesos
- Sincronización

Comunicación entre procesos

- Relación entre procesos
 - Independientes: no puede afectar, ni ser afectado por los demás procesos que se ejecutan en el sistema, compiten por el uso de recursos escasos
 - Cooperativos: puede afectar o ser afectado por los demás procesos que se ejecutan en el sistema, colaboran entre sí buscando un objetivo común.

Comunicación entre procesos

- IPC: Inter Processes Communication
 - Compartir espacios de memoria,
 - Paso de mensajes entre procesos
- Tipos:
 - Síncrona vs asíncrona
 - Persistente vs volátil

Comunicación entre procesos

- Comunicación entre procesos
 - RPC: remote procedure call
 - Desde el punto de vista de un programador l funciona de la misma manera que si la llamada fuese local : transparente
 - Cliente – Servidor
 - Mensajes
 - Asíncrono
 - Servidor puede no estar disponible

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- **Gestión de procesos**
- Sincronización

Gestión de procesos

- Windows
 - Ctr + Alt + Sup
- Unix
 - Comando *top*
 - Lista los procesos en ejecución
 - Comando kill:
 - *kill -9 pid*
 - -9: fuerza a que el SO mate el proceso

Programación multiproceso

- Definiciones
- Procesos
- Servicios
- Hilos
- Programación concurrente
- Programación paralela
- Programación distribuida
- Creación de procesos
- Comunicación entre procesos
- Gestión de procesos
- **Sincronización**

Sincronización entre procesos

- El Sistema Operativo se encarga de enviarle señales a los procesos para coordinar su evolución y conseguir que progrese armónicamente

Sincronización entre procesos

- Sección crítica: código que en un instante solo puede ser ejecutado por un único proceso o hilo
- La concurrencia de varios procesos/hilos en una sección crítica dejaría el programa en un estado inestable
- Se debe implementar la exclusión mutua

Sincronización entre procesos

- Exclusión mutua
 - Solo un proceso por sección crítica y recurso compartido.
 - El mismo proceso no puede usar el mismo recurso compartido indefinidamente.
 - Todo proceso debe entrar antes o después a usar el recurso.
 - Si el recurso está sin uso, cualquiera podrá disponer de él inmediatamente.
 - Si hay varios esperando usar el recurso y se libera, uno de ellos lo usará durante un tiempo determinado

Sincronización entre procesos

- Java: *synchronized*
- Utilizado en:
 - Métodos de clase
 - Métodos estáticos
 - Bloques de código
- Se debe sincronizar la menor parte de código posible: penaliza el rendimiento

Sincronización entre procesos

- Métodos synchronized

```
private synchronized void instanceMethod() {  
    //do stuff  
}  
  
private static synchronized void staticMethod() {  
    //do stuff  
}
```

Sincronización entre procesos

- Java: synchronized block
 - synchronized(this): sincroniza el acceso al código para la instancia actual

```
private void mehod() {  
    //do stuff  
    synchronized (this) {  
        //do more stuff  
        //...  
    }  
}
```

Programación multiproceso

- Documentación
 - Wikipedia
 - Facultad informática UVA
 - <http://cidecame.uaeh.edu.mx>