Universidad Nacional Autónoma de México

Los procesos y su creación

Proyecto PAPIME PE104911

Pertinencia de la enseñanza del cómputo paralelo en el currículo de las ingenierías.

Luis García
Proyecto PAPIME PE104911

Intel para la Academia
Facultad de Ingeniería, UNAM

Proceso

- ¿Qué es un proceso?

Cuando un programa es leído del disco por el kernel y es cargado en memoria para ejecutarse, se convierte en PROCESO

Procesos en UNIX

- Un proceso en Unix está conformado por tres bloques fundamentales.

- Segmento de Texto
- Segmento de Datos
- Segmento de Pila

Segmento de Texto

- Contiene las instrucciones que entiende la CPU de nuestra máquina. Este bloque es una copia del bloque de texto del programa.

Segmento de Datos

- Contiene los datos que deben ser inicializados al arrancar el proceso.

Si usamos el compilador de C, en este bloque 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
- Éste está conformado por marcos de pila, sirve para funciones, variables así como información útil.

Estados de un Proceso en UNIX

1. El proceso se está ejecutando en modo usuario
2. El proceso se está ejecutando en modo supervisor
3. El proceso no se está ejecutando, pero está listo para ejecutarse tan pronto como el Scheduler lo ordene
4. El proceso está durmiendo cargado en memoria

Estados de un Proceso en UNIX

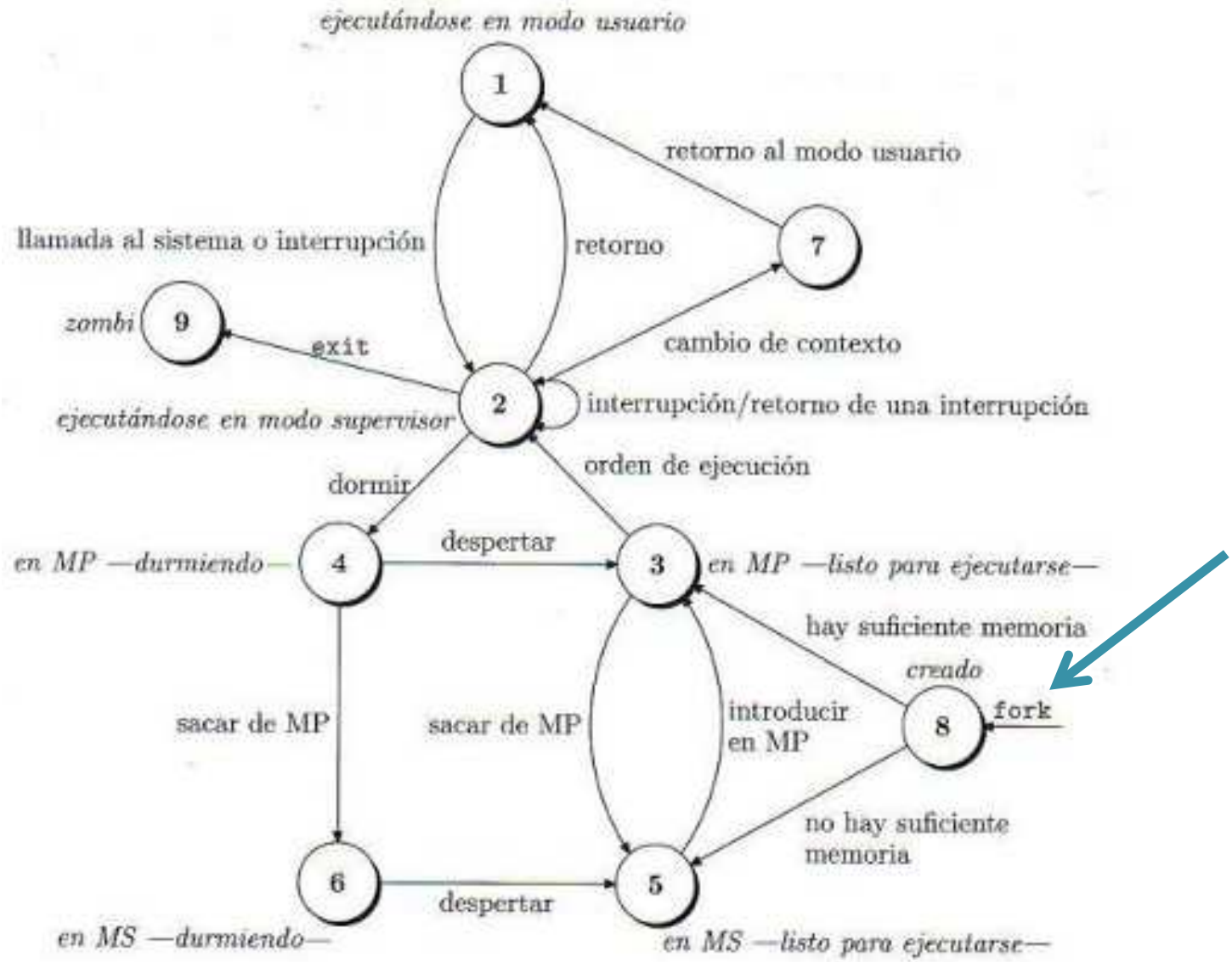
5. El proceso está listo para ejecutarse, pero el intercambiador --Proceso 0-- debe cargar el proceso en memoria antes de que el Scheduler pueda ordenar que pase a ejecutarse
6. El proceso está durmiendo y el intercambiador ha descargado el proceso hacia una memoria secundaria --swap-- para crear espacio en la memoria principal donde poder cargar otros procesos.

Estados de un Proceso en UNIX

7. El proceso está volviendo del modo supervisor al modo usuario, pero el núcleo se apropia del proceso y hace un cambio de contexto, pasando otro proceso a ejecutarse en modo usuario
8. El proceso acaba de ser creado y está en un estado de transición; el proceso existe, pero ni está preparado para ejecutarse --Estado 3--, ni durmiendo –Estado 4--. Este estado es el inicial para todos los procesos, excepto el proceso 0 --Init--.

Estados de un Proceso en UNIX

9. El proceso ejecuta la llamada exit y pasa al estado zombi. El proceso ya no existe, pero deja para su proceso padre un registro que contiene el código de salida y algunos datos estadísticos tales como los tiempos de ejecución. El estado de zombi es el estado final de un proceso.



Luis García
Proyecto PAPIME PE104911

Intel para la Academia
Facultad de Ingeniería, UNAM

Creación de procesos (fork)

```
#include <sys/types.h>
pid_t fork ();
```

```
int pid;
…
If( ( pid = fork( ) ) == -1)
    perror (“Error en la llamada a fork”);
else  if (pid==0)
            // Código que ejecutará el proceso hijo
else
            // Código que ejecutará el proceso padre
```

Luis García
Proyecto PAPIME PE104911

Intel para la Academia
Facultad de Ingeniería, UNAM

Terminación de procesos (exit y wait)

- Exit

```
#include <stdlib.h>
void exit (int status);
```

- Wait

```
#include <sys/types.h>
#include <sys/wait.h>
pid_t wait (int *stat_loc);
```

Información sobre procesos

- Identificadores de proceso

```
#include <types.h>
pid_t getpid ( );
pid_t getppid ( );

#include <sys/types.h>
pid_t getpgrp ( );
pid_t setpgrp ( );
```

Identificadores sobre procesos

- Identificadores de usuario y de grupo

```
#include <sys/types.h>
uid_t getuid ( );
uid_t geteuid ( );
gid_t getgid ( );
gid_t getegid ( );

int setuid (uid_t uid);
int setgid (gid_t gid);
```