Unidad Didáctica 1. Arranque y procesos del sistema

Alvaro Vazquez Vazquez

September 30, 2025

I.E.S. Fernando Aguilar Quignon
C/Conil de la Frontera, 3
CP 11010, Cádiz

Administración de Sistemas Operativos - 1ª Evaluación (RA 2 – CE a, b, c)

A continuación se van a exponer una serie de preguntas relacionadas con el punto 1 y 2 de la presente unidad. Deberán ser respondidas de manera extensa, utilizando el lenguaje técnico aprendido en clase.

Pregunta 1. Explica qué es un programa y un proceso, realizando una comparación de ambos.
Respuesta: Un proceso es un programa en ejecucion, que el sistema operativo carga y gestiona en memoria para que la CPU pueda ejecutarlo. Un programa es un conjunto de instrucciones escritas en un lenguaje de programación que realiza una tarea específica cuando se ejecuta. Un programa es un archivo estático almacenado en el disco, mientras que un proceso es una instancia dinámica de ese programa en ejecución. Un programa puede tener múltiples procesos asociados, cada uno con su propio estado y contexto de ejecución.

Pregunta 2. ¿Qué elementos contiene una imagen de proceso? Define cada uno de ellos.
Respuesta:

(a) Bloque de control de proceso (BCP): El BCP es una estructura de datos que mantiene el sistema operativo. Contiene informacion sobre el proceso. Como punteros, estado del proceso, numero de proceso, contador de programa, registros, espacio de direcciones, listas de archivos abiertos y datos de contabilidad y estado.

(b) Instrucciones del programa en codigo maquina. Son las funciones del codigo, en machine code, como assembly.

(c) Datos estaticos del programa(constantes y variables.)

(d) Pilas de usuario y kernel (llamada a funciones.)

(e) Heap o monticulo (datos creados de forma dinamica)

Pregunta 3. ¿Qué estructura utiliza el sistema operativo para gestionar, de forma global, los procesos que están corriendo?
Respuesta: Con la tabla de procesos del sistema el SO tiene una estructura que permite localizar el bloque de control de un determinado proceso revisando el puntero en la tabla.

Pregunta 4. ¿Qué elementos componen el BCP?
Respuesta: La estructura del BCP contiene:

(a) Puntero
(b) Estado del proceso
(c) Numero de proceso
(d) Contador de programa
(e) Registros
(f) Espacio de direcciones
(g) Lista de archivos abiertos
(h) Datos de contabilidad

Pregunta 5. ¿Para qué usa el sistema el CP?
Respuesta: El CP o contador de programa almacena la direccion de memoria de la siguiente instruccion a ejecutar por la CPU

Pregunta 6. ¿Por qué es útil almacenar el estado de los registros de la CPU?
Respuesta: Cuando un proceso con mayor prioridad entra en la CPU "cambio de contexto" se debe almacenar el estado del proceso anterior para que, cuando se reanude su ejecucion, la CPU puede continuar exactamente por donde lo dejo.

Pregunta 7. ¿Qué implica que un proceso corra en modo kernel? ¿Qué ocurrirá si se ocasiona un error durante dicha ejecución?
Respuesta: Cuando un proceso corre en modo kernel, significa que tiene acceso sin restricciones al hardware, en especial a la memoria principal (RAM). Todo codigo que se ejecute en modo kernel compartira un unico espacio de memoria. Si ocurre algun error durante dicha ejecucion, el proceso podria escribir en una zona de memoria critica o erronea, bloqueando el sistema (kernel panic)

Pregunta 8. Relacionado con lo anterior, ¿cuál es el mensaje típico que muestra el sistema?
Respuesta: Si un driver (proceso que es comun que corra en modo kernel) se bloquea, bloquearia todo el sistema, causando un kernel panic.

Pregunta 9. ¿Para qué sirven las llamadas a sistema?
Respuesta: Los procesos que se ejecutan en modo usuario, no tiene acceso directo al hardware, disponen de un subconjunto de instrucciones del numero total de instrucciones que ofrece la CPU. Por ello, este modo de ejecucion cuando requiera algun recurso hardware, necesitara hacer una **llamada a sistema**, que le permite solicitar al sistema operativo la operación requerida de forma segura

Pregunta 10. En relación con el espacio de direcciones virtuales que puede utilizar un proceso, ¿qué diferencia existe entre el modo usuario y el modo kernel?
Respuesta: Los procesos que se ejecutan en modo kernel tienen acceso a un espacio de direcciones compartido con el sistema operativo y otros procesos en modo kernel, mientras que los procesos que se ejecutan en modo usuario tienen un espacio de direcciones privado aislado de otros procesos y del kernel, lo que garantiza seguridad y estabilidad del sistema.

Pregunta 11. ¿Qué implicaciones tiene el hecho de que un proceso se esté ejecutando en primer plano? ¿Y en segundo?
Respuesta:

- Los procesos que se ejecutan en primer plano son aquellos que necesitan que un usuario los inicie o interactue con ellos. Por ejemplo, si hemos iniciado el proceso desde la terminal, este la va a ocupar, y vamos a tener que esperar a que finalice para poder iniciar uno nuevo.
- Por otro lado los procesos en segundo plano, no necesitamos que finalicen para poder utilizar la terminal. Son procesos que corren en el sistema sin interaccion con el usuario.
  Para ejecutar uno escribiremos el comando seguido de un ampersand.

```
1        firefox &
```

Pregunta 12. ¿Qué términos anglosajones utilizamos para referirnos a los procesos que corren en primer plano? ¿Y en segundo?
Respuesta:

- Los procesos en primer plano se denominan foreground procesos
- Mientras que los procesos en segundo plano se denominan background process

Pregunta 13. ¿Qué principal diferencia existe entre un trabajo y un daemon?
Respuesta:

Todos los daemons son procesos que se ejecutan en segundo plano, pero no todos los procesos en segundo plano son daemons. Un daemon es un proceso que se ejecuta en segundo plano y que no está asociado a una terminal o sesión de usuario específica. Los daemons suelen iniciarse durante el arranque del sistema y continúan ejecutándose hasta que el sistema se apaga. Estos procesos suelen encargarse de tareas del sistema, como la gestión de servicios de red, la impresión, la programación de tareas, entre otros.

Pregunta 14. ¿En qué plano corre un daemon? ¿Y un trabajo?
Respuesta:

Un daemon corre en segundo plano, mientras que un trabajo puede correr en primer o segundo plano. Un trabajo es cualquier programa que ejecutemos vinculado a una terminal de forma interactiva sin desasociarse de la misma. Este trabajo se compondrá de una serie de tareas.

Pregunta 15. ¿Cómo pasamos un proceso que está corriendo en primer plano a segundo plano?
Respuesta: Para pasar de un proceso en primer plano a segundo plano debemos primero utilizar el atajo de teclado Ctrl + z. Este atajo detendrá el proceso por lo que no estará trabajando para nosotros, sin embargo, dejará nuestra terminal libre para que la podamos utilizar. Para reanudar el proceso, simplemente debemos ejecutar el comando jobs, ver en qué posición se encuentra el proceso que queramos manipular, en este caso, reanudarlo y que continúe en segundo plano. Y ejecutar el siguiente comando bg (porcentaje)1 . Siendo 1 la posición del proceso al ejecutar jobs.

Nota bg "background" lo reanuda en segundo plano, fg o "foreground" lo reanuda en primer plano.

Pregunta 16. ¿Cómo se llama la señal que hemos utilizado para llevar a cabo el proceso anterior?
Respuesta:

Señal SIGSTOP que envía el proceso a segundo plano y lo detiene. En la actividad anterior he utilizado el atajo de teclado Ctrl + z pero en realidad el comando es "kill -SIGSTOP -pid-"

Pregunta 17. Cuando examinamos el estado de los procesos vía terminal, ¿qué símbolo tienen los daemons en el apartado de terminal (TT)?
Respuesta:

Tienen el símbolo de interrogación, quiere decir que no tienen asociada ninguna terminal.

Pregunta 18. ¿Qué eventos principales provocan la creación de procesos?
Respuesta:

- Arranque del sistema.
- Ejecucion de un proceso, llamada para creacion de otro proceso.
- Peticion de un usuario.
- Inicio de trabajo por lotes.

Pregunta 19. ¿Qué ocurre cuando el algoritmo de planificación asociado a una CPU es de tipo apropiativo? ¿Y no apropiativo?
Respuesta:

- Cuando es un algoritmo apropiativo, un proceso, por ejemplo, de mayor prioridad puede forzar un cambio de contexto para entrar en la CPU, interrumpiendo el proceso de menor prioridad.
- En cambio los algoritmos de tipo no asociativos no permiten que haya un cambio de contexto hasta que el proceso que haya empezado a ejecutarse no haya terminado. Esto a veces no es lo conveniente debido a que se pueden formar convoyes de procesos.

Pregunta 20. Explica los estados relacionados con el modelo de 5 estados.
Respuesta:

A lo largo de la ejecución de un proceso, este pasará por diversos estados según la disponibilidad dentro del sistema operativo. Estos cambios vienen motivados debido a la naturaleza de los procesadores, que solo pueden ejecutar un proceso simultáneamente.

Los estados de los procesos según el algoritmo de 5 estados son:

- Nuevo: Acaba de ser creado y está esperando su admisión por parte del Sistema operativo.
- Listo o preparado: Está preparado para su ejecución una vez ha sido admitido. A la espera de que se le asigne uso de CPU.
- En ejecución: Como el mismo término indica, ya está ejecutándose dentro de la CPU, consumiendo recursos.
- Bloqueado: El proceso está esperando que se produzca un determinado evento (p.e, que se realice una operación de lectura o escritura que él mismo haya solicitado.) Cuando un proceso está en este estado, libera los recursos de la CPU y no compite por su uso.
- Terminado: El proceso ha finalizado, pero la imagen del mismo no ha sido eliminada de memoria.

Pregunta 21. Dentro del ámbito de la gestión de procesos, ¿qué planificadores existen? ¿Cuál es su cometido?
Respuesta:

Existen planificadores a corto, medio y largo plazo.

Pregunta 22. ¿Qué relación existe entre la memoria virtual y la gestión de procesos?
Respuesta:

La memoria virtual permite que los procesos utilicen más memoria de la que físicamente está disponible en el sistema. Esto es útil cuando se ejecutan múltiples procesos simultáneamente, ya que cada uno puede requerir una cantidad significativa de memoria. La memoria virtual ayuda a gestionar estos requisitos de memoria al permitir que los procesos utilicen espacio en disco como si fuera memoria RAM, lo que mejora la eficiencia y la capacidad del sistema para manejar múltiples procesos.

Pregunta 23. ¿Qué estados se añaden en el modelo de 7 estados?
Respuesta:

Se añaden los estados de Suspendido listo y Suspendido bloqueado.

Pregunta 24. Bajo qué circunstancias y desde qué estados se alcanzan los nuevos estados?
Respuesta:

- El estado de suspendido listo se alcanza desde el estado de listo, cuando el sistema operativo decide liberar memoria, moviendo el proceso a memoria secundaria (swap). El proceso permanece en este estado hasta que se le vuelva a asignar memoria principal y pueda volver al estado de listo.
- El estado de suspendido bloqueado se alcanza desde el estado de bloqueado, cuando el sistema operativo decide liberar memoria, moviendo el proceso a memoria secundaria (swap). El proceso permanece en este estado hasta que se le vuelva a asignar memoria principal y pueda volver al estado de bloqueado.

Pregunta 25. ¿Qué dos señales del sistema están relacionadas con los anteriores estados?
Respuesta:

- SIGSTOP: Detiene el proceso, enviándolo a segundo plano.
- SIGTSTP: Detiene el proceso, enviándolo a segundo plano y liberando la terminal.

Pregunta 26. Dentro de los sistemas GNU/Linux, ¿qué simbolizan los estados S, D, R y T?
Respuesta:

- S -> Sleeping: El proceso está en ejecución pero está a la espera de que otro evento termine. No está ejecutando ninguna instrucción en la CPU.
- D -> Bloqueado: Igual que lo explicado anteriormente en la actividad 20. El proceso está esperando que se produzca un determinado evento (p.e, que se realice una operación de lectura o escritura que él mismo haya solicitado.) Cuando un proceso está en este estado, libera los recursos de la CPU y no compite por su uso.

Pregunta 27. En el caso del formato BSD, ¿qué significan los modificadores <, N, s, l y +?
Respuesta:

- < : Significa que es un proceso de alta prioridad, por lo que en algoritmos de planificación apropiativos, podrían forzar un cambio de contexto con un proceso de menor prioridad.
- N : Es un proceso de baja prioridad
- s : Implica que el proceso es el líder de sesión.
- + : Proceso en primer plano
- l : Proceso que está utilizando páginas de memoria bloqueadas en RAM.

Pregunta 28. Define trabajo, tarea y proceso.
Respuesta:

Un trabajo es cualquier programa que ejecutemos vinculado a una terminal de forma interactiva sin desasociarse de la misma. Este trabajo se compondrá de una serie de tareas.

Estas tareas son consideradas como una subparte de un trabajo, combinando varias tareas podremos completar un trabajo determinado. En este caso podríamos decir que una tarea es un subproceso o hilo de ejecución de un trabajo.

Un proceso es un programa en ejecucion, que el sistema operativo carga en memoria para que pueda ser ejecutado por la CPU. Un proceso puede componerse de varias tareas o subprocesos, y a su vez un trabajo se compone de varias tareas.

Pregunta 29. ¿Qué relación existe entre una tarea y un trabajo?
Respuesta:

Como ya he mencionado anteriormente un trabajo se compone de un conjunto de tareas, que se ejecutarán secuencialmente para lograr un objetivo. Un trabajo se puede identificar como un solo proceso, que se podrá dividir en varias tareas, las cuales se pueden identificar como un conjunto de subprocesos o hilos.

Pregunta 30. ¿Una tarea puede ser un trabajo? ¿Y viceversa?
Respuesta:

Una tarea no puede considerarse un trabajo, ya que un trabajo se compone de varias tareas. Sin embargo, un trabajo si puede considerarse una tarea, ya que es un conjunto de tareas.

Pregunta 31. ¿Por qué decimos que una tarea se puede considerar un subproceso?
Respuesta:

Por que un conjunto de tareas componen un trabajo, un trabajo se puede identificar como un solo proceso y un proceso se puede dividir en varios subprocesos o hilos. Por lo tanto una tarea se puede considerar como un subproceso.

Pregunta 32. ¿Qué diferencia existe entre un trabajo y un proceso?
Respuesta:

Un proceso puede ejecutarse en primer o segundo plano. Es decir puede ejecutarse estando o no asociado a una shell de forma interactiva. Mientras que un trabajo se caracteriza por ser un programa iniciado de forma interactiva, asociado a una shell. Por tanto, todos los trabajos son procesos pero no todos los procesos son trabajos.

Pregunta 33. ¿Cuándo consideramos que un proceso es un trabajo?
Respuesta:

Un proceso es un trabajo cuando está vinculado e iniciado bajo una shell o terminal.

Pregunta 34. En relación con las terminales, ¿qué ámbito o contexto tienen los trabajos?
Respuesta:

Los trabajos tienen un ámbito o contexto de usuario, ya que son iniciados por un usuario de forma interactiva. Por tanto, los trabajos no pueden ser iniciados por el sistema operativo de forma automática, como si ocurre con los daemons.

Pregunta 35. ¿De qué dos maneras, vía terminal, podemos revisar los trabajos?
Respuesta:

- Ejecutando el comando "jobs"
- o ejecutando el comando "ps aux" y revisar el campo TTY.

Pregunta 36. ¿Qué es un hilo de ejecución?
Respuesta:

Un hilo de ejecución es la unidad más pequeña de procesamiento que puede ser programada por un sistema operativo. Es una secuencia de instrucciones dentro de un proceso que puede ejecutarse de manera independiente. Un proceso puede contener múltiples hilos, los cuales comparten el mismo espacio de direcciones y recursos del proceso padre, pero cada hilo tiene su propio contador de programa, pila y registros.

Pregunta 37. ¿Qué concepto está relacionado con dichos hilos?
Respuesta:

La concurrencia y la paralelismo. La concurrencia se refiere a la capacidad de un sistema para manejar múltiples tareas al mismo tiempo, mientras que el paralelismo implica la ejecución simultánea de múltiples tareas en diferentes núcleos de procesador. Los hilos permiten que un proceso realice múltiples tareas de manera concurrente, mejorando la eficiencia y el rendimiento del sistema (explicado en ejercicios posteriores). (Tambien podriamos hablar de hyperthreading)

Pregunta 38. ¿Qué problema se deriva de la no existencia de programación multihilo?
Respuesta:

La ausencia de programación multihilo provoca una utilización ineficiente de los recursos del sistema, ya que un proceso debe ejecutar sus tareas de forma secuencial. Esto puede causar bloqueos, tiempos de respuesta más lentos y menor capacidad para manejar múltiples solicitudes simultáneamente. Además, impide aprovechar al completo los procesadores multinúcleo, lo que afecta al rendimiento de las aplicaciones modernas y reduce su capacidad de respuesta en entornos concurrentes.

Pregunta 39. ¿Qué ocurre con la imagen del proceso cuando tenemos varios hilos corriendo del mismo proceso?
Respuesta:

Los recursos y la memoria asignados al proceso padre, son compartidos entre todos los hilos del mismo proceso. Todos los hilos comparten la misma imagen del proceso.

Pregunta 40. ¿Por qué los hilos aumentan la eficiencia de la comunicación entre programas que están en ejecución?

Respuesta:

Debido a que los hilos comparten el mismo espacio de direcciones y recursos del proceso padre, la comunicación entre hilos es más rápida y eficiente que la comunicación entre procesos separados. Los hilos pueden intercambiar datos directamente a través de variables compartidas en memoria. Esto se traduce en:

- Creacion mas eficiente de nuevos hilos a partir de un proceso exactamente, que crear un proceso nuevo.
- Menor tiempo en la terminacion de un hilo que en la terminacion de un proceso.
- Menor tiempo en un cambio de contexto entre hilos que entre procesos.
- Mayor eficiencia en la comunicacion entre hilos debido a que comparten el mismo espacio de direcciones y recursos del proceso padre.

Pregunta 41. ¿Por qué es más liviano el cambio de contexto entre dos hilos de un mismo proceso?

Respuesta:

Porque los hilos comparten el mismo espacio de direcciones y recursos del proceso padre, por lo que no es necesario cambiar el espacio de direcciones ni los recursos asignados al proceso padre. Imaginemos que tenemos dos hilos A y B del mismo proceso. Si el hilo A está en ejecución y se produce un cambio de contexto para que el hilo B pueda ejecutarse, solo es necesario guardar el estado del hilo A (contador de programa, registros y pila) y cargar el estado del hilo B. No es necesario cambiar el espacio de direcciones ni los recursos asignados al proceso padre.

Pregunta 42. Define el concepto de interrupción.

Respuesta:

Una interrupción es una señal recibida por la CPU que indica que debe interrumpir la ejecucion actual y pasar a ejecutar codigo especifico que trate una determinada situacion.

Pregunta 43. Define el concepto de excepción.

Respuesta:

Lsd excepciones son un tipo de interrupción que emplea el procesador para notificar al sistema operativo cuando se produce un evento inesperado durante la ejecucion de un programa, como una division por cero o un acceso a memoria invalido.

Pregunta 44. ¿Cuándo se da una interrupción por software? ¿En qué consisten?

Respuesta:

Son generadas por el propio programa en ejecucion, mediante una instruccion especial llamada llamada a sistema. Se producen cuando un proceso en modo usuario necesita acceder a un recurso del sistema operativo, como la lectura o escritura de de disco. El proceso se interrumpe virtualmente hasta recibir respuesta, es decir, hasta que se complete la operación solicitada. Durante esta interrupcion, las instrucciones ejecutadas no seran del proceso sino del propio sistema operativo. Una vez finalizada la operacion, el proceso se reanudara desde el punto donde fue interrumpido.

Pregunta 45. ¿Cuándo se da una interrupción por hardware? ¿En qué consisten?
Respuesta:

En general, las interrupciones por hardware son el resultado de operaciones de E/S. No son pruducto de un programa, sino las señales enviadas por los dispositivos perifericos para indicarle al procesador que requieren su atencion.

Pregunta 46. ¿Cuál fue la primera técnica que se implementó para las interrupciones?
Respuesta:

La primera tecnica fue la de sondeo o polling.

Pregunta 47. ¿Cómo se aplicaba la técnica anterior?
Respuesta:

Consistia en que el procesador revisaba de forma periodica el estado de los dispositivos perifericos para comprobar si alguno de ellos necesitaba su atencion. Si un dispositivo necesitaba atencion, el procesador interrumpia la ejecucion del programa en curso y atendia la solicitud del dispositivo. Esta tecnica era ineficiente, ya que el procesador perdia tiempo revisando el estado de los dispositivos perifericos en lugar de ejecutar instrucciones del programa en curso. Esto no puede admitirse en sistemas multitarea.

Pregunta 48. ¿Qué ocurre con los modos de ejecución cuando se lanza una excepción por software?
Respuesta:

Cuando se lanza una excepcion por software, se genera un cambio de contexto, pasando del modo usuario al modo kernel o modo supervisor para que el sistema operativo pueda atender el error.

Pregunta 49. ¿Por qué decimos que las excepciones son un mecanismo de protección que permite garantizar la integridad de los datos almacenados tanto en el espacio del usuario como en el espacio del kernel?
Respuesta:

Porque las excepciones permiten al sistema operativo detectar y manejar errores o situaciones anormales que puedan comprometer la integridad de los datos. Al interceptar estas excepciones, el sistema operativo puede tomar medidas para evitar que los procesos en modo usuario accedan a áreas de memoria críticas del sistema o realicen operaciones no autorizadas

Pregunta 50. ¿Bajo qué circunstancias se produce una excepción?
Respuesta:

- Division por cero
- Acceso a memoria invalido
- Instruccion ilegal
- Desbordamiento de pila
- Violacion de proteccion