***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) Unidad Didáctica 1. Arranque y procesos del sistema

**1. Explica qué es un programa y un proceso, realizando una comparación de ambos.**

| **PROGRAMA** | **PROCESO** |
| --- | --- |
| Es estático, sólo es el código escrito | Está en ejecución y usa recursos del sistema |
| Se guarda en dispositivos de almacenamiento como SSD o Discos duros | Se carga en la memoria RAM |
| Sigue existiendo hasta que se elimine del disco | Solo existe mientras se esté ejecutando |

**2. ¿Qué elementos contiene una imagen de proceso? Define cada uno de ellos.**

* **Código del programa** → Instrucciones del programa que están siendo ejecutadas
* **Datos estáticos**: Datos variables globales y constantes definidos antes de la ejecución de un proceso
* **Datos dinámicos:** Memoria asignada durante la ejecución.
* **Pila**: Contiene variables locales y datos de funciones.
* **Contador de programa**: Dirección de la próxima instrucción a ejecutar.
* **Registros de la CPU**: Almacenan temporalmente datos y resultados intermedios
* **Archivos abiertos**: Información de los archivos usados por el proceso.
* **Memoria compartida**: Área para la comunicación con otros procesos.
* **Información de E/S**: Recursos de entrada/salida usados por el proceso

**3. ¿Qué estructura utiliza el sistema operativo para gestionar, de forma global, los procesos que están corriendo?**

El sistema operativo utiliza una estructura llamada tabla de control de procesos. El BCP es necesario para el cambio de contexto, que es el entorno de ejecución de un proceso.

**4. ¿Qué elementos componen el BCP?**

* Puntero.
* Estado del proceso.
* Número de proceso.
* Contador de programa.
* Registros.
* Espacio de direcciones. Relacionado con la gestión de memoria.
* Lista de archivos abiertos. Incluye la lista de archivos abiertos por el proceso.
* Datos de contabilidad y estado.

**5. ¿Para qué usa el sistema el CP?**

Indica la dirección de memoria de la próxima instrucción que el procesador debe ejecutar

**6. ¿Por qué es útil almacenar el estado de los registros de la CPU?**

Permite al procesador interrumpir la tarea para poder continuarla en el mismo punto donde la dejó.

**7. ¿Qué implica que un proceso corra en modo kernel? ¿Qué ocurrirá si se ocasiona un error durante dicha ejecución?**

Un proceso que corre en modo kernel tiene acceso completo a los recursos del sistema, si se produce un error durante la ejecución el impacto es mayor y desencadena fallos como el kernel panic. Pero por tema de eficiencia es más rápido ya que puede consultar con el sistema operativo en cualquier momento.

**8. Relacionado con lo anterior, ¿cuál es el mensaje típico que muestra el sistema?**

En Linux: Kernel Panic y en Windows: Pantallazo azul.

**9. ¿Para qué sirven las llamadas al sistema?**

Permiten a los procesos en modo usuario solicitar servicios del sistema operativo

**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?**

| **MODO USUARIO** | **MODO KERNEL** |
| --- | --- |
| Los procesos tienen acceso a un espacio de direcciones limitado. Este espacio está controlado por el sistema operativo | El sistema operativo tiene acceso completo al espacio de direcciones y a todos los recursos de la máquina |

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

| **PRIMER PLANO** | **SEGUNDO PLANO** |
| --- | --- |
| Son aquellos que necesitan que un usuario los inicie o que interactúe con ellos. | No deberemos esperar a su finalización para poder interactuar con la terminal.  Son procesos que están corriendo en el sistema sin la atención e interacción del usuario. |

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

* **Primer plano**: Foreground
* **Segundo plano**: Background

**13. ¿Qué diferencia principal existe entre un trabajo y un daemon?**

| **TRABAJO** | **DAEMON** |
| --- | --- |
| Puede ejecutarse en primer o segundo plano e interactuar con el usuario en la terminal | Es un proceso del sistema que se ejecuta en segundo plano de manera continua, generalmente sin intervención del usuario |

**14. ¿En qué plano corre un daemon? ¿Y un trabajo?**

* **Daemon**: Siempre corre en segundo plano
* **Trabajo**: Puede ejecutarse tanto en primer plano como en segundo plano

**15. ¿Cómo pasamos un proceso que está corriendo en primer plano a segundo plano?**

Primero lo suspendemos usando **CTRL + Z** y luego usamos el comando **bg** para reanudarlo

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

La señal utilizada para suspender el proceso es SIGTSTP

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

Los daemons normalmente tienen el símbolo “**?”** en el apartado de terminal, indicando que no están asociados a una terminal específica.

**18. ¿Qué eventos principales provocan la creación de procesos?**

* Arranque del sistema.
* Ejecución de un proceso, llamada para la creación de otro proceso.
* Petición del usuario.
* Inicio de trabajo por lotes.

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

* **Apropiativos:** El sistema operativo puede interrumpir un proceso en ejecución para asignar la CPU a otro proceso de mayor prioridad
* **No apropiativo:** Se asigna la CPU por un periodo de tiempo, sin la posibilidad de que otro proceso le arrebate su uso.

**20. Explica los estados relacionados con el modelo de 5 estados.**

* **Nuevo:** El proceso ha sido creado, pero aún no ha comenzado su ejecución.
* **Listo:** El proceso está preparado para ejecutarse y espera que la CPU esté disponible.
* **En ejecución**
* **Bloqueado:** El proceso no puede continuar hasta que se resuelva un evento externo
* **Terminado.** El proceso ha finalizado, pero su imagen aún no ha sido eliminada de la memoria.

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

* **Planificador a largo plazo:** Determina qué procesos se admiten en el sistema.
* **Planificador a corto plazo:** Es el encargado de seleccionar qué proceso, de entre los que están en estado listo, obtendrá la CPU para ejecutarse
* **Planificador a medio plazo:** Realiza la suspensión y reanudación de procesos, moviéndolos entre la memoria y el almacenamiento secundario

**22. ¿Qué relación existe entre la memoria virtual y la gestión de procesos?**

**23. ¿Qué estados se añaden en el modelo de 7 estados?**

Se añaden los estados de Suspendido bloqueado y Suspendido liso

**24. ¿Bajo qué circunstancias y desde qué estados se alcanzan los nuevos estados?**

**25. ¿Qué dos señales del sistema están relacionadas con los anteriores estados?**

Están relacionadas SIGSTOP y SIGCONT.

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

* **S**: Durmiendo. El proceso está en ejecución, pero en ese momento, no se encuentra ejecutandose ninguna acción
* **D**: Bloqueado. El proceso está esperando la finalización de una operación externa y no puede ser interrumpido hasta que la operación se complete
* **R**: En ejecución. El proceso se está ejecutando de forma activa en la CPU
* **T**: Parado por la señal SIGSTOP

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

* **<.** Alta prioridad.
* **N.** Baja prioridad**.**
* **L**. Páginas bloqueadas en memoria.
* **s.** Líder de sesión. Se da cuando el PID == SID.
* **+.** Proceso en foreground.

**28. Define trabajo, tarea y proceso.**

* **Proceso**: Conjunto de acciones o comandos enviados al sistema operativo, que puede incluir varios procesos o tareas.
* **Trabajo:** Son programas asociados siempre a la shell.
* **Tareas:** Es la manifestación de un programa que el sistema operativo gestiona para que se ejecute.

**29. ¿Qué relación existe entre una tarea y un trabajo?**

Un trabajo puede estar compuesto por una o varias tareas

**30. ¿Una tarea puede ser un trabajo? ¿Y viceversa?**

Una tarea puede ser un trabajo cuando el trabajo consiste en una única acción o proceso. Y un trabajo puede estar compuesto por múltiples tareas

**31. ¿Por qué decimos que una tarea se puede considerar un subproceso?**

Porque una tarea puede dividirse en múltiples subprocesos que comparten los mismos recursos, pero ejecutan diferentes partes del código en paralelo, logrando mayor eficiencia

**32. ¿Qué diferencia existe entre un trabajo y un proceso?**

* Un trabajo es una acción completa que se solicita al sistema operativo y puede involucrar múltiples procesos
* Un proceso es una instancia específica de un programa en ejecución

**33. ¿Cuándo consideramos que un proceso es un trabajo?**

Un proceso se considera un trabajo cuando el proceso representa toda la tarea o acción solicitada al sistema operativo

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

Los trabajos pueden ejecutarse en primer plano o en segundo plano, dependiendo de si interactúan directamente con el usuario o se ejecutan en segundo plano sin intervención.

**35. ¿De qué dos maneras, vía terminal, podemos revisar los trabajos?**

Podemos usar el comando jobs y el comando ps

**36. ¿Qué es un hilo de ejecución?**

Unidad de procesamiento dentro de un proceso que ejecuta instrucciones de forma independiente.

**37. ¿Qué concepto está relacionado con dichos hilos?**

Multihilo

**38. ¿Qué problema se deriva de la no existencia de programación multihilo?**

La ejecución secuencial reduce el rendimiento y aprovecha menos los procesadores multinúcleo.

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

Los hilos comparten los mismos recursos del proceso pero gestionan sus propias secuencias de ejecución de manera independiente

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

Los hilos dentro del mismo proceso comparten el mismo espacio de memoria, lo que permite que la comunicación entre ellos sea más rápida y eficiente

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

Porque los hilos comparten la misma memoria y recursos, así el sistema no necesita cambiar la imagen completa del proceso

**42. Define el concepto de interrupción.**

Señal que pausa la ejecución del procesador para atender un evento urgente

**43. Define el concepto de excepción.**

Evento anómalo durante la ejecución de un programa que requiere intervención del sistema.

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

Una interrupción por software se produce cuando un programa genera una señal para el procesador, solicitando la atención del sistema operativo

**45. ¿Cuándo se da una interrupción por hardware? ¿En qué consisten?**

Una interrupción por hardware ocurre cuando un dispositivo externo envía una señal al procesador para indicar que necesita atención. El procesador pausa su ejecución para atender la solicitud del hardware.

**46. ¿Cuál fue la primera técnica que se implementó para las interrupciones?**

Fue el uso del polin.

**47. ¿Cómo se aplicaba la técnica anterior?**

Se asignaba una dirección de memoria específica a cada tipo de interrupción para saber cómo manejarla.

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

El procesador cambia del modo usuario al modo kernel para permitir que el sistema operativo gestione la excepción de manera segura y adecuada, garantizando que se mantenga el control sobre los recursos del sistema

**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?**

Las excepciones permiten al sistema operativo intervenir y corregir errores de los programas, evitando que operaciones incorrectas afecten el funcionamiento del sistema o comprometan la integridad de los datos.

**50. ¿Bajo qué circunstancias se produce una excepción?**

Las excepciones se producen cuando ocurre un evento inesperado durante la ejecución de un programa, como una instrucción inválida