



Universidad de Murcia
Facultad de Informática



Universidad Politécnica de Cartagena
Escuela Técnica Superior de Ingeniería de
Telecomunicación

TÍTULO DE GRADO EN
CIENCIA E INGENIERÍA DE DATOS
Fundamentos de Computadores

Tema 4: El sistema operativo

Boletín de autoevaluación de teoría / problemas

CURSO 2023 / 24

Índice general

I. Ejercicios resueltos	2
II. Ejercicios propuestos	6
III. Soluciones a los ejercicios resueltos	7

Ejercicios resueltos

1. Señala cuál de las siguientes afirmaciones sobre el sistema operativo Linux es correcta:
 - a) Diseñado inicialmente como alternativa a Windows en los procesadores Intel, actualmente corre exclusivamente sobre éstos.
 - b) Su código fuente es abierto.
 - c) Utiliza únicamente su propio sistema de ficheros, por lo que el manejo de ficheros de otros sistemas operativos se hace complicado.
2. A pesar de sus diferencias, las distintas distribuciones de Linux comparten siempre:
 - a) Un mismo sistema de archivos (salvo quizá diferencia de versión).
 - b) Un mismo entorno gráfico (salvo quizá diferencia de versión).
 - c) Un mismo núcleo del SO (salvo quizá diferencia de versión).
3. ¿Cuál de las siguientes sería la definición más correcta de proceso?
 - a) El código que ejecuta el procesador en un periodo de tiempo durante el cual el usuario interactúa con el sistema.
 - b) Un código que se ejecuta cuando estamos en modo núcleo.
 - c) Un programa en ejecución.
4. En Linux, todos los procesos son descendientes (directos o no) del proceso llamado ____, que es el primer proceso iniciado por el ____ al arrancar el sistema. Los procesos pueden terminar de modo voluntario, ejecutando la llamada al sistema ____, o “matados” desde el exterior, usando la llamada ____.
5. De las siguientes afirmaciones, señala cual es falsa:
 - a) Es el planificador de Linux el que se encarga de gobernar el cambio de tarea.
 - b) Linux permite la multitarea, siempre y cuando existan varios procesadores o núcleos en la CPU.
 - c) Los procesos bloqueados en una E/S provocan la cesión de la CPU a otro proceso.
6. A la parte software del sistema operativo que controla un dispositivo de E/S se le denomina ____ del dispositivo, mientras que al chip que controla físicamente dicho dispositivo se le denomina ____ del dispositivo.
7. Una de las mayores ventajas del uso del intérprete de comandos queda evidenciada porque:
 - a) Se pueden combinar comandos para hacer cosas muy potentes.
 - b) Es la única manera de comprimir ficheros para que ocupen menor tamaño en disco.
 - c) No hay otra manera de cambiar los permisos a un directorio o fichero.
8. El sistema operativo de un PC:
 - a) Es una parte del hardware (almacenada en ROM) que toma el control al arrancar el ordenador.
 - b) Es una capa software que toma el control al arrancar el ordenador, y administra el entorno de ejecución de programas.
 - c) Es un programa que sirve únicamente para inicializar el ordenador y que, al cargar el entorno gráfico, termina su ejecución.

9. Terminología usada en sistemas operativos: A un programa en ejecución se le denomina habitualmente ____, mientras que al periodo de tiempo durante el cual un usuario interactúa con el sistema se le llama _____. Aquellos sistemas operativos que permiten su uso simultáneo por varios usuarios se denominan sistemas operativos _____. Para interactuar con un sistema operativo, se puede hacer a través de un GUI, siglas que significan exactamente ____ (en castellano ____), o bien tecleando órdenes en un terminal, método denominado interfaz de _____.
10. Explica brevemente la labor del planificador (*scheduler*) de un sistema operativo, y qué relación tiene con las interrupciones a la CPU por parte del reloj del sistema.
11. Explicar así mismo qué hace el *scheduler* en el momento en que a) llega una petición de E/S por parte de un proceso, y b) cuando el dispositivo de E/S termina la transacción.
12. Un *driver* de dispositivo:
- a) Ejecuta código que se comunica con la controladora del dispositivo.
 - b) Es un chip que controla físicamente el dispositivo.
 - c) Es siempre un proceso de usuario independiente del *kernel*.
13. POSIX es:
- a) Un estándar para definir interfaces de llamadas al sistema.
 - b) Un estándar de tratamiento de interrupciones software.
 - c) Un estándar de tratamiento de interrupciones hardware.
14. De las siguientes afirmaciones, señala cual es falsa:
- a) Cada proceso del sistema tiene su propia memoria virtual, que el SO puede mapear a distintas direcciones de memoria física.
 - b) Todos los procesos del sistema administran directamente su propia memoria física.
 - c) Parte del espacio de memoria virtual de un proceso puede estar en disco.
15. En Linux, los planificadores de procesos y memoria, y los drivers de E/S se ejecutan:
- a) Como una cierta parte (subrutina) del proceso *ps*.
 - b) En el modo usuario de la CPU.
 - c) En el modo núcleo de la CPU.
16. A las llamadas al sistema por parte de un proceso también se las conoce como:
- a) Interrupciones software.
 - b) Interrupciones hardware.
 - c) Excepciones.
17. Los ficheros, la memoria virtual y los procesos son:
- a) Aplicaciones que corren encima del sistema operativo.
 - b) Abstracciones que ofrece el sistema operativo a los usuarios.
 - c) Parte del hardware y, por tanto, son dependientes de la microarquitectura del procesador.
 - d) Una conjunción de procesador + memoria + E/S.
18. En el ámbito de los sistemas operativos, los cambios de contexto permiten:

- a) Cambios en el proceso que se encuentra en ejecución.
 - b) La apertura y cierre de aplicaciones de forma segura.
 - c) Las llamadas al sistema.
 - d) El cambio en los registros y espacio de memoria usada por los procesos.
19. La memoria virtual que ofrece el sistema operativo es:
- a) Más grande que la cantidad de memoria disponible en el disco duro.
 - b) Más grande que la cantidad de memoria disponible en la memoria RAM.
 - c) Siempre la misma para cada sistema, e igual a la memoria RAM disponible.
 - d) Igual a la cantidad de memoria disponible en la memoria RAM más la cantidad de memoria caché.
20. Justifica si la siguiente afirmación es correcta, sabiendo que tanto Linux como MacOS son sistemas UNIX que soportan POSIX: si mi programa está escrito según el IEEE POSIX y lo compilo para que se ejecute en mi ordenador con SO Linux Ubuntu, el fichero ejecutable (binario) servirá igualmente para un ordenador con sistema operativo MacOS.
21. Los programas de nivel de usuario (como un procesador de textos) pueden acceder directamente al espacio en disco para guardar un fichero.
- a) Verdadero, pues pueden abrir y cerrar ficheros directamente para modificar su contenido.
 - b) Falso, porque solamente pueden modificar datos de la memoria RAM.
 - c) Verdadero, ya que se pueden ejecutar en modo núcleo.
 - d) Falso, pues el acceso a los ficheros lo gestiona el sistema operativo en modo núcleo.
22. ¿Qué entidad software del sistema operativo decide qué proceso ejecutar en cada instante en un sistema multi-tarea?
- a) El planificador de procesos.
 - b) La CPU.
 - c) El sistema de ficheros.
 - d) El gestor de contextos.
23. ¿Qué elemento de un computador se encarga de la gestión de la memoria por cada uno de los programas que se lanzan a ejecución?
- a) El gestor de memoria virtual.
 - b) La CPU.
 - c) La CPU y el sistema operativo.
 - d) El sistema operativo.
24. Si nuestro sistema operativo Linux usa un tamaño de página de 64 KB, responde a las siguientes preguntas:
- a) ¿A qué cantidad de páginas físicas (marcos) podríamos mapear páginas virtuales, si nuestra CPU usa 39 bits para las direcciones físicas?
 - b) ¿Qué cantidad de páginas virtuales tendrían disponibles las aplicaciones si nuestra CPU sigue el ISA x86-64, con 64 bits para las direcciones virtuales?
25. Si el tamaño del conjunto de trabajo de páginas virtuales de los programas en ejecución es mayor al tamaño de la memoria principal:

- a) Sucede lo que se denomina como excepción por fallo de página.
- b) Se aumenta el tamaño del espacio de paginación en el disco duro y mejora el rendimiento.
- c) El rendimiento del sistema se ve mermado por la paginación con el disco duro.
- d) El sistema operativo empieza a cerrar procesos.

26. El tamaño de un directorio en un sistema UNIX viene dado por:

- a) El número de entradas que hay en la lista de enlaces a ficheros.
- b) La suma del tamaño de los archivos que están dentro.
- c) La longitud del nombre.
- d) La longitud del nombre junto con las entradas para los directorios . (punto) y .. (punto punto).

27. Un *driver* de un teclado es:

- a) Un componente hardware que sirve para que el SO pueda manejar el teclado.
- b) Un módulo software que se encarga de comunicarse con la controladora del teclado.
- c) Una parte del núcleo del sistema operativo.
- d) Una estructura de datos que guarda información sobre el fichero que representa al dispositivo de E/S (teclado).

28. ¿Qué es la Unidad de Gestión de Memoria (MMU)?:

- a) Un elemento hardware que ayuda al sistema operativo a gestionar el acceso a las páginas de memoria virtual.
- b) Un módulo hardware que contiene un pequeño número de traducciones de página virtual a página física.
- c) Es una memoria caché.
- d) Es una parte del sistema operativo que siempre está en ejecución para ayudar a la traducción de direcciones de memoria virtual.

29. ¿Dónde se guardan los datos relativos a los ficheros abiertos en el sistema?:

- a) En la tabla de ficheros abiertos.
- b) En la tabla de descriptores.
- c) En cada descriptor de fichero.
- d) En los bloques de datos del fichero.

Ejercicios propuestos

1. Define brevemente, indicando las similitudes y diferencias entre ellas, los conceptos de interrupción, excepción y *trap*.
2. Explica brevemente, apoyándote con un ejemplo, el mecanismo que usa el sistema operativo para gestionar una llamada al sistema que involucre una entrada/salida con un dispositivo. ¿Tiene esto algo que ver con la multitarea? ¿Y con las interrupciones? Razona brevemente tus respuestas.
3. Explica brevemente qué es el redireccionamiento al ejecutar un proceso.
4. Justifica si un sistema operativo Linux actual, como Ubuntu, incluye un interfaz de usuario de tipo gráfico o de línea de comandos.
5. Explica el proceso de creación de un proceso y los estados por los que puede pasar éste.
6. Describe brevemente cómo funciona el sistema de gestión de la memoria virtual del sistema operativo mediante paginación.
7. Justifica por qué la gestión de la memoria virtual y de la caché del procesador siguen un comportamiento similar y su gestión es equivalente, aunque a diferentes niveles.
8. Describe la ventaja de usar ficheros para representar dispositivos en los sistemas UNIX.
9. Describir para qué sirven los descriptores de fichero en un sistema operativo.
10. Las llamadas al sistema se pueden considerar:
 - a) Excepciones asíncronas.
 - b) Interrupciones hardware.
 - c) Cambios de contexto del proceso en ejecución.
 - d) Excepciones síncronas.
11. El espacio de memoria de los procesos contiene las siguientes partes:
 - a) Pila, montón, datos y código.
 - b) Datos, código, pila y variables globales.
 - c) Datos, instrucciones del núcleo y pila de ejecución.
 - d) Pila, datos, código y registros.
12. ¿Cuál es el proceso *init* en los sistemas operativos tipo UNIX?:
 - a) El proceso de inicio del entorno gráfico del sistema operativo.
 - b) No es un proceso como tal, sino un espacio especial de memoria del sistema operativo.
 - c) Es un proceso de apertura del procesador de comandos.
 - d) Es el primer proceso que se lanza en el sistema.
13. Explica cómo ayudan la MMU y el TLB al sistema operativo en la gestión de la memoria.

Soluciones a los ejercicios resueltos

1. *Solución:* b)
2. *Solución:* c)
3. *Solución:* c)
4. *Solución:* `init`, sistema operativo, `exit`, `kill`.
5. *Solución:* b)
6. *Solución:* Driver, controladora.
7. *Solución:* a)
8. *Solución:* b)
9. *Solución:* Proceso, sesión, multiusuario, Graphical User Interface, Interfaz Gráfico de Usuario, línea de comandos.
10. El *scheduler* permite el uso compartido de la(s) CPU(s) entre distintos procesos. Para ello, periódicamente (un elevado número de veces por segundo), el reloj del sistema interrumpe al proceso actual, y entra a ejecutarse código del núcleo (*kernel*), que cambia el contexto para pasar a ejecutar otro proceso. Así, todos los procesos van avanzando con sensación de simultaneidad, a pesar de que en realidad se están ejecutando por turnos (aunque, en todo caso, también puede haber paralelismo real si el procesador dispone de varios núcleos).
11. Los procesos que realizan una llamada al sistema para realizar una operación de E/S (p.e., lectura de disco) quedan temporalmente bloqueados, y el *scheduler* cede entonces automáticamente la CPU a otro(s) proceso(s). Mientras dicha transacción no ha terminado, el planificador mantiene en ejecución en la CPU a otros procesos que no están bloqueados en una E/S (por turnos, como se aclaraba en la solución anterior), siempre intentando optimizar el uso de este recurso. Finalmente, cuando el dispositivo de E/S ha terminado su tarea, el proceso que realizó la llamada al sistema es desbloqueado, y pasa de nuevo a “competir” con el resto de procesos no bloqueados por la CPU.
12. *Solución:* a)
13. *Solución:* a)
14. *Solución:* b)
15. *Solución:* c)
16. *Solución:* a)
17. *Solución:* b)
18. *Solución:* a)
19. *Solución:* b)
20. Si un programa está escrito haciendo uso de librerías compatibles con POSIX en Linux, esto garantiza que puede funcionar para otro sistema operativo que sea compatible con POSIX, como es el caso de MacOS. No obstante, los ficheros ejecutables que se generan en un proceso de compilación (traducción) para cada sistema operativo son distintos y, por tanto, no se pueden usar directamente en otro sistema operativo. Sería necesario compilar (traducir) el programa para MacOS.

21. *Solución:* d)
22. *Solución:* a)
23. *Solución:* c)
24. La solución viene dada por los siguientes cálculos, según hemos visto en clase:
- a) Con 39 bits para las direcciones físicas ($P = 39$), y un tamaño de página de 64 KB (2^{16}), tendremos $2^{39}/2^{16} = 2^{23}$ marcos.
 - b) Con 64 bits para las direcciones virtuales ($V = 64$), cada proceso tendrá como máximo $2^{64}/2^{16} = 2^{38}$ páginas virtuales.
25. *Solución:* c)
26. *Solución:* a)
27. *Solución:* b)
28. *Solución:* a)
29. *Solución:* a)