

Repaso

Segundo Parcial Teórico

Memoria (2da parte), E/S, Archivos, Buffer Cache

Memoria

1. La paginación por demanda es una forma de implementar memoria virtual.
2. En la administración segmentada de memoria hay una tabla general para todo el sistema.
3. Para solucionar un page fault, habrá que hacer en algún momento un context switch.
4. Si un proceso contara con los frames que necesita, no provocaría ningún page fault (falla de página).
5. La tabla de páginas es parte de: el contexto / la PCB / el espacio de direcciones / la pila del proceso
6. El tamaño de la tabla de segmentos depende del tamaño del proceso.
7. Si disminuyo la cantidad de bits del desplazamiento de una dirección de memoria, los frames serán más chicos.
8. La cantidad máxima de páginas en memoria depende sólo del tamaño del proceso.
9. EL bit de modificación activo relacionado con una página generará una I/O adicional.
10. Analice tamaños de página y page fault.
11. Relación de tamaño de página, de proceso, de tabla de páginas según la arquitectura de la dirección.
12. La tabla invertida proporciona acceso directo al marco buscado.
13. ¿Qué es el working set? Es siempre el mismo?
14. ¿Qué consecuencias puede tener la hiperpaginación?
15. En la técnica del Conjunto Trabajo ¿Qué pasa si el delta elegido es muy chico? ¿Y si es muy grande?
16. Diferencia entre reemplazo de páginas global y local.
17. Con el reemplazo local no cambia la cantidad de frames asignados al proceso
18. Diferencia entre asignación equitativa y proporcional.
19. Secuencia de resolución de page fault incluyendo TLB.
20. En cuanto a los estados del bit M y R: ¿Cuál sería la página ideal para elegir como pagina victima?

Archivos

21. En qué momento se hace el chequeo sobre si el usuario puede acceder a un archivo: en el open? en cada read? en cada write?
22. En Unix System V: ¿puede modificarse el i-nodo del archivo sin modificar el archivo en sí?
23. En Unix System V: ¿puede modificarse el archivo sin modificar su i-nodo?

24. En Unix System V se verán beneficiados en performance los archivos cuyo contenido pueda ser referenciado por las 10 primeras direcciones de bloque que están en su i-nodo.
25. En Unix System V, el acceso random a un archivo puede realizarse accediendo directamente al bloque que necesito, sin leer los precedentes.
26. En Unix System V, Puede asignarse un bloque a un archivo sin acceder previamente al superblock?
27. En Unix System V, Se puede acceder a un archivo sin acceder a su i-nodo.
28. En Unix System V, al crear un archivo en un filesystem, indique qué se modifica:
-directorio al que pertenece -superblock -lista de i-nodos de todos los filesystems.
29. En Unix System V, puedo crear un archivo en un filesystem no montado?
30. Todos los filesystems de un disco deben tener el mismo tamaño de bloque.
31. Cuando un archivo se borra, se ponen en cero los bloques
32. La estructura del filesystem define el tamaño máximo del archivo.
33. La estructura del filesystem define la longitud máxima del nombre

Buffer Cache

34. En la estructura de Buffer cache vista, un buffer puede estar ocupado y delayed write a la vez
35. El inodo de un archivo que se está usando debe estar en algún buffer del buffer cache.
36. Un buffer delayed write puede volver a estar ocupado, si lo pide un proceso, pero antes debe grabarse a disco. (importa el bloque en el buffer)
37. Para asignar un bloque a un archivo es necesario contar con el superblock en el buffer cache.
38. Un proceso esperando por un buffer delayed write y ocupado, deberá esperar la escritura a disco antes de que se le asigne a él. (importa el bloque en el buffer)
39. Las hash queues sirven para buscar por un buffer en particular
40. La free list sirve para buscar por cualquier buffer.
41. No puede haber más de un proceso esperando por un buffer.
42. Cuando un proceso libera un delayed write, es escrito a disco antes de ponerlo en la free list.
43. Puede un buffer en la free list, estar ocupado?
44. Si un buffer está primero en la free list y en ese momento lo pide un proceso: donde va cuando se libere?
45. Si un proceso necesita un buffer y la free list está vacía, el proceso se aborta.
46. Si una hash queue está vacía, se toma un buffer de otra cola.
47. Para acceder a la tabla de páginas, ésta debe estar completa en el buffer cache.
48. Que conviene mas? mas cantidad de hash queues con pocos elementos, o menos cantidad de hash queues con mas elementos.