

Repaso

Segundo Parcial Teórico

Memoria, E/S, Archivos, Buffer Cache

Memoria

1. En paginación, si disminuyo la cantidad de bits del desplazamiento de una dirección de memoria, los frames serán más chicos.

Verdadero

2. La cantidad máxima de páginas en memoria depende sólo del tamaño del proceso.

No, de otras cosas mas (criterios) xd

3. Analice tamaños de página y page fault.

Mayo y menor meh, la cosa es cuando estan en un tamaño intermedio

4. Relación de tamaño de página, de proceso, de tabla de páginas según la arquitectura de la dirección.

Mucho texto

5. La tabla invertida proporciona acceso directo al marco buscado.

si

6. Qué consecuencias puede tener la hiperpaginación.

Perdida de perfomance

7. En la técnica del Conjunto Trabajo ¿Qué pasa si el delta elegido es muy chico?

¿Y si es muy grande?

Delta chico, puedo quedarme con pocas paginas

Delta muy grande me pueden sobrar demasiadas

8. Diferencia entre reemplazo de páginas global y local.

Global, le saca a otros procesos y agrande el working set

Local, solo saca del proceso mismo

9. Con el reemplazo local no cambia la cantidad de frames asignados al proceso

verdadero

10. Diferencia entre asignación equitativa y proporcional.

Lo mismo para todos vs depende de la demanda de cada uno

11. Secuencia de resolución de un page fault (incluyendo TLB)

Tlb, tabla de paginas, page fault, cambiar tlb

12. En cuanto a los estados del bit M y R: ¿Cuál sería la página ideal para elegir como pagina victima?

0 y 0, pero tiene prioridad el bit R

## Archivos

13. 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?

En el open, despues ya no

14. En Unix System V: ¿puede modificarse el i-nodo del archivo sin modificar el contenido del archivo en sí?

si

15. En Unix System V: ¿puede modificarse el contenido del archivo sin modificar su i-nodo?

No, se cambia la fecha

16. 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.

si

17. En Unix System V, el acceso random a un archivo puede realizarse accediendo directamente al bloque que necesito, sin leer los precedentes.

Si, con calculo

18. En Unix System V, Puede asignarse un bloque a un archivo sin acceder previamente al superblock?

No, porque esta el mapa de bit que tiene el estado de almacenamiento

19. En Unix System V, Se puede acceder a un archivo sin acceder a su i-nodo.

No, tiene las claves de todo

20. En Unix System V, al crear un archivo en un filesystem, indique qué se modifica:

-directorio al que pertenece -superblock -tabla de i-nodos

Los tres xd

21. En Unix System V, puedo crear un archivo en un filesystem no montado?

NO

22. Todos los filesystems de un disco deben tener el mismo tamaño de bloque.

No necesariamente (claster) bloque del disco sector si es unico tama

23. Cuando un archivo se borra, se ponen en cero los bloques?

no

24. La estructura del filesystem define el tamaño máximo del archivo.

si

25. La estructura del filesystem define la longitud máxima del nombre

si

## Buffer Cache

26. En la estructura de Buffer cache vista, un buffer puede estar ocupado y delayed write a la vez si

27. El inodo de un archivo que se está usando debe estar en algún buffer del buffer cache.

No, porque en cache solo hay bloques de datos. Los inodos son bloques de control

28. 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)

No necesariamente imprimir, da igual

29. Para asignar un bloque a un archivo es necesario contar con el superblock en el buffer cache.

No, porque es otra cosa no bloque de datos

30. 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)

no

31. Las hash queues sirven para buscar por un bloque o buffer en particular

Solo por bloque

32. La free list sirve para buscar por cualquier buffer.

Solo los buffer libres

33. No puede haber más de un proceso esperando por un buffer.

Puede haber mas de uno

34. Cuando un proceso libera un delayed write, es escrito a disco antes de ponerlo en la free list.

No tiene nada que ver

35. Puede un buffer en la free list, estar ocupado?

NO

36. Si un buffer está primero en la free list y en ese momento lo pide un proceso:

donde va cuando se libere?

Al final de la lista

37. Si un proceso necesita un buffer y la free list está vacía, el proceso se aborta.

No, espera

38. Si una hash queue está vacía, se toma un buffer de otra cola.

Si no esta se carga, se busca en la free list y puede estar en otra hash queue

39. Para acceder a la tabla de páginas, ésta debe estar completa en el buffer cache.

No, no es de datos

40. Que conviene mas? mas cantidad de hash queues con pocos elementos, o menos cantidad de hash queues con mas elementos.

Muchos y pocos elementos