

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

File Systems 1

Sistemas Operativos y Distribuidos

Prof. Javier Echaiz
D.C.I.C. – U.N.S.
<http://cs.uns.edu.ar/~jechaiz>
je@cs.uns.edu.ar




1

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Concepto de Archivo

- Espacio de direcciones lógicas contiguas
- Tipos:
 - Datos
 - numérico
 - caracter
 - binario
 - Programa

2

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura de Archivo

- Ninguna – secuencia de palabras, bytes
- Estructura de registros simple
 - Líneas
 - Longitud fija
 - Longitud variable
- Estructuras Complejas
 - Documento con formato
 - Archivo de carga reubicable
- Se pueden simular estos dos últimos puntos con el primer método por la inserción de caracteres de control apropiados.
- Lo decide:
 - El sistema operativo
 - El programa

3

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Atributos de Archivo

- ▶ **Nombre** – mantiene información en forma legible.
- ▶ **Tipo** – necesario para sistemas que soportan diferentes tipos.
- ▶ **Ubicación** – puntero a la ubicación del archivo en el dispositivo.
- ▶ **Tamaño** – tamaño actual del archivo.
- ▶ **Protección** – controla quien puede leer, escribir, ejecutar.
- ▶ **Tiempo, fecha, e identificación de usuario** – datos para protección, seguridad, visualización de uso.
- ▶ La información sobre los archivos es mantenida en la estructura de directorio, la que es mantenida en el disco.

4

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Operaciones sobre Archivos

- creación
- escritura
- lectura
- reposición puntero actual (*seek*)
- borrado
- truncado
- $open(F_i)$ – busca la estructura de directorio en el disco para la entrada F_i , y mueve el contenido de la entrada a la memoria.
- $close(F_i)$ – mueve el contenido de la entrada F_i en la memoria a la estructura del directorio en el disco.

5

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Archivos Abiertos

- Son necesarios varios datos para administrar los archivos abiertos:
 - Puntero corriente del archivo: punteros a la última ubicación read/write, hay un puntero por proceso que tiene el archivo abierto.
 - Cuenta de archivo abierto: cuenta el número de veces que el archivo es abierto, permite remover datos de la tabla de archivos abiertos cuando el último proceso lo cierra.
 - Ubicación en el disco del archivo: información de acceso a datos en el caché.
 - Permisos de acceso: información del modo de acceso por proceso.

6

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Tipos de Archivo – nombre, extensión

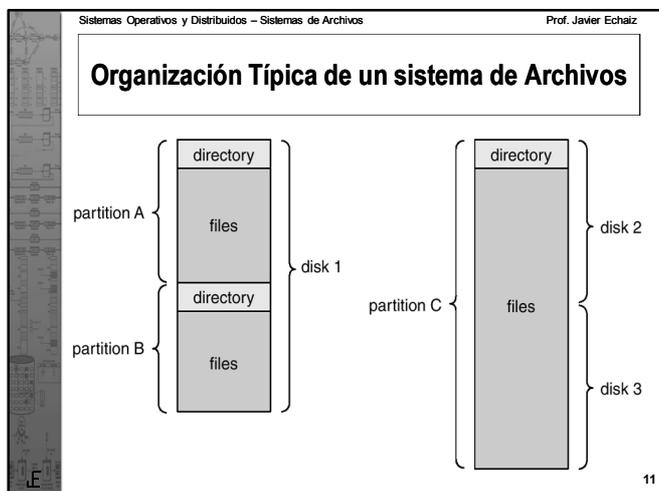
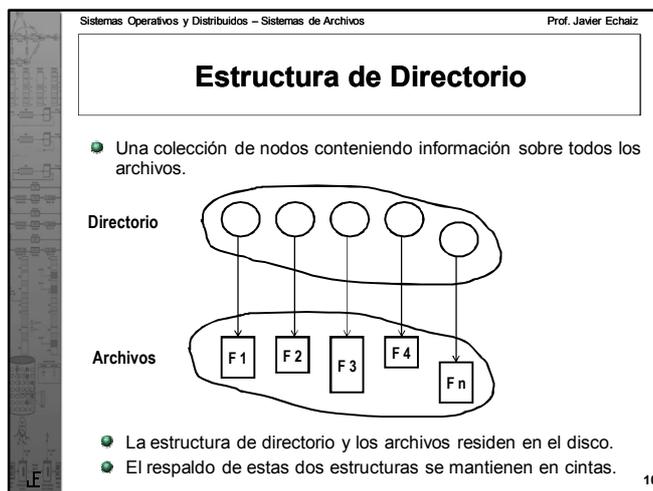
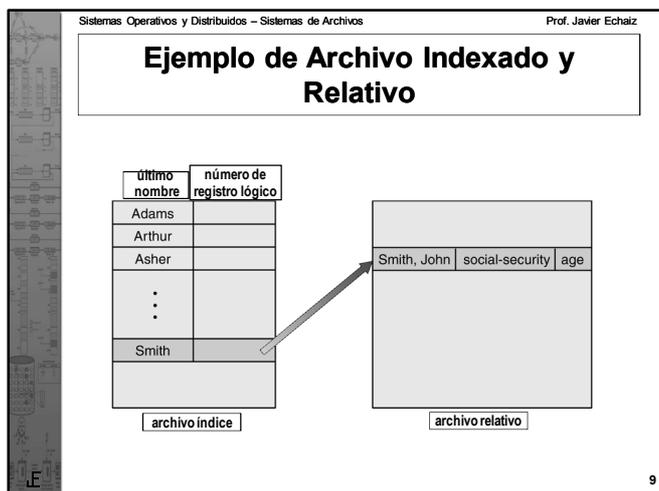
| Tipo de Archivo | Extensión usual | Función |
|-----------------|-------------------------|--|
| Ejecutable | exe, com, bin o ninguno | programa leng. máquina listo para correr |
| Objeto | obj, o | compilado, leng máquina, no enlazado |
| Código fuente | c, p, pas, l77,asm, a | código fuente en varios lenguajes |
| Lote (batch) | bat, sh | comandos al intérprete de comandos |
| Texto | txt, asc | doc con datos textuales |
| Procesa. texto | docx, tex, rtf | formatos de proc de palab |
| Librería | lib, a | librerías de rutinas |
| Imp o vista | ps, dvi, gif | ASCII o archivos binarios |
| Archivo | rar, bz2, zip, tar, 7z | arch relacionados agrupados en un solo archivo |
| Multimedia | mpeg, mov, rm, jpg | arch binarios conteniendo audio o info AV |

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Métodos de Accesos

- Acceso Secuencial**
 - read next
 - write next
 - reset
 - no read after last write (rewrite)
- Acceso Directo**
 - read n
 - write n
 - position to n
 - read next
 - write next
 - rewrite n

n = número relativo de bloque



- Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz
- ### Información en un Directorio de Dispositivo
- Nombre
 - Tipo
 - Dirección
 - Longitud corriente
 - Máxima longitud
 - Fecha del último acceso
 - Fecha de la última actualización (para dump)
 - ID del dueño
 - Información de protección

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Operaciones sobre un Directorio

- Búsqueda de un archivo
- Creación de un archivo
- Borrado de un archivo
- Listado de un directorio
- Renombrado de un archivo
- Recorrer un sistema de archivos

13

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Objetivos de Organización

- Eficiencia – localizar un archivo rápidamente.
- Nombres – conveniente para los usuarios.
 - ♦ Dos usuarios pueden tener el mismo nombre para diferentes archivos.
 - ♦ El mismo archivo puede tener varios nombres diferentes.
- Agrupamiento – agrupamiento lógico de archivos por propiedades, (p.e., todos los programas C#, todos los juegos, ...)

14

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura Arbórea de Directorios

15

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura Arbórea de Directorios (Cont.)

- Búsqueda eficiente
- Capacidad de agrupamiento
- Directorio actual (directorio de trabajo)
 - ♦ `cd /spell/mail/prog`
 - ♦ `cat list`
- ♦ Probar el siguiente alias:


```
♦ dtree='find ./ -type d -print | sed -e "s:[^/]*/;|____;g;s;____; |;g"'
```

16

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura Arbórea de Directorios (Cont.)

- Camino (path) de nombres absoluto o relativo
- La creación de un nuevo archivo se hace en el directorio corriente.
- Borrado de un archivo


```
rm <nombre-arch>
```
- La creación de un nuevo subdirectorio se hace en el directorio corriente.


```
mkdir <nombre-dir>
```

Ejemplo: si el directorio corriente es /spell/mail

```
mkdir count => /spell/mail/count
```
- Borrar "mail" => borrar el subárbol entero cuya raíz es "mail".

17

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Archivos Compartidos

- Es deseable compartir archivos en un sistema multiusuario.
- La acción de compartir debe ser hecha por medio de un esquema de *protección*.
- En sistemas distribuidos los archivos pueden ser compartidos a través de la red.
- Network File System (NFS) es un método común de compartir archivos distribuidos.

18

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Protección

- El **creador/duño** del archivo debería poder controlar:
 - que cosas pueden hacerse
 - por quién
- Tipos de acceso
 - Read
 - Write
 - Execute
 - Append
 - Delete
 - List

19

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Listas de Acceso y Grupos

- Modos de acceso: read, write, execute
- Tres clases de usuarios
 - RWX
 - a) acceso dueño 7 ⇒ 1 1 1
 - b) acceso grupo 6 ⇒ 1 1 0
 - c) acceso público 1 ⇒ 0 0 1
- Pedir al administrador crear un grupo (único nombre), sea G, y adicionar algún usuario al mismo.
- Para un archivo particular (sea *game*) o subdirectorio, definir un acceso apropiado.

```

                dueño  grupo  público
                 /   |   \
                chmod 761 game
                 /   |   \
                chgrp  G  game
            
```

- Cambiar un grupo a un archivo

20

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Semántica de Consistencia

- ¿Qué es consistencia?
- UNIX
 - Las escrituras son inmediatamente visibles
 - Pueden compartir el puntero de ubicación
- Sesión (Andrew)
 - Las escrituras no son inmediatamente visibles
 - Los cambios son visibles después que el archivo es cerrado.

21

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Manejo de Listas de Acceso en Windows

| Permissions for Administrator | Allow | Deny |
|-------------------------------|--------------------------|--------------------------|
| Full Control | <input type="checkbox"/> | <input type="checkbox"/> |
| Modify | <input type="checkbox"/> | <input type="checkbox"/> |
| Read & Execute | <input type="checkbox"/> | <input type="checkbox"/> |
| Read | <input type="checkbox"/> | <input type="checkbox"/> |
| Write | <input type="checkbox"/> | <input type="checkbox"/> |
| Special Permissions | <input type="checkbox"/> | <input type="checkbox"/> |

22

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Ejemplo de Listado de Directorio en UNIX

```

drwxr-xr-x  9 je je      4096 2009-11-15 23:04 .
drwxr-xr-x 15 je root    4096 2009-11-09 15:21 ..
-rw-r----- 1 je je      720896 2009-09-06 18:56 00-Admin.ppt
-rw-r----- 1 je je      644096 2009-09-06 23:54 01-Intro.ppt
-rw-r----- 1 je je     1422336 2009-09-09 00:21 02-Estructuras.ppt
-rw-r----- 1 je je     1295872 2009-10-07 00:47 03-Procesos.ppt
-rw-r----- 1 je je      685056 2009-10-26 00:51 04-SincronizacionExtras.ppt
-rw-r----- 1 je je      765952 2009-11-01 22:53 04-Sincronizacion.ppt
-rw-r----- 1 je je      304128 2009-10-26 00:46 04-SincronizacionProbClasicos.ppt
-rw-r----- 1 je je      443904 2009-11-01 22:58 05-Deadlocks-extra.doc
-rw-r----- 1 je je      438784 2009-11-01 23:09 05-Deadlocks.ppt
-rw-r----- 1 je je     1020416 2009-11-04 03:00 06-GestionMemoria-DSM.ppt
-rw-r----- 1 je je      782848 2009-11-04 03:08 06-GestionMemoria-Extras.ppt
-rw-r----- 1 je je     1389056 2009-11-15 23:04 07-FileSystems.ppt
-rw-r----- 1 je je      546816 2009-09-06 18:56 base.ppt
-rw-r----- 1 je je     274944 2009-09-27 16:42 base_sosd.ppt
drwxr-xr-x  2 je je      4096 2009-09-08 21:53 books
    
```

23

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura del Sistema de Archivos

- Estructura de Archivo
 - Unidad Lógica de almacenamiento
 - Colección de información relacionada
- El sistema de archivos reside en almacenamiento secundario (discos).
- El sistema de archivo está organizado en capas.
- *File Control Block* – estructura de almacenamiento consistente de información sobre el archivo.

24

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura del Sistema de Archivos

Un FCB típico

| |
|------------------------------------|
| file permissions |
| file dates (create, access, write) |
| file owner, group, ACL |
| file size |
| file data blocks |

25

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en General

26

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructuras de Archivo

- Bloques Físicos
- Registros Lógicos
- Fragmentación

Bloques físicos

Registros lógicos

27

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructuras del Sistema de Archivos en Memoria

a) Apertura de un archivo b) Lectura de un archivo

28

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Métodos de Asignación

Un método de asignación se refiere a cómo los bloques de disco de un archivo son ubicados:

- Asignación Contigua
- Asignación Enlazada
- Asignación Indexada

29

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Contigua

- Cada archivo ocupa un conjunto de bloques contiguos en el disco.
- Simple – solo se necesita la ubicación de comienzo (block #) y la longitud (número del bloques).
- Acceso aleatorio.
- Desperdicio de espacio (problema de asignación dinámica).
- Los archivos no pueden crecer.
- Mapeo de lógico a físico.

$$\text{Dirección Lógica/512} \begin{cases} Q \\ R \end{cases}$$

Bloque a ser accedido = Q + dirección de comienzo
Desplazamiento dentro del bloque = R

30

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Contigua

| file | start | length |
|-------|-------|--------|
| count | 0 | 2 |
| tr | 14 | 3 |
| mail | 19 | 6 |
| list | 28 | 4 |
| f | 6 | 2 |

31

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Contigua

- El archivo es definido por base y longitud
- Soporta el acceso secuencial y directo

- Dificultad para asignar el espacio o incrementar el tamaño del archivo

32

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Enlazada

- Cada archivo es una lista enlazada de bloques de disco: los bloques pueden estar en cualquier lugar del disco.

| file | start | end |
|------|-------|-----|
| jeep | 9 | 25 |

33

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Enlazada

- El archivo es definido por primero y último
- Resuelve el problema de almacenamiento - cualquier bloque libre servirá
- No soporta (eficientemente) el acceso directo
- Simple – necesita solo la dirección inicial
- Sistema de administración del espacio libre – no malgasta espacio
- No hay acceso aleatorio

34

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Enlazada (Cont.)

■ **File-Allocation Table (FAT)** – asignación de espacio de disco usado en MS-DOS y OS/2.

| entrada de directorio | nombre | bloque inicial |
|-----------------------|--------|----------------|
| test | | 217 |

| Nº de bloques del disco - 1 | valor |
|-----------------------------|-------|
| 0 | 618 |
| 339 | eof |
| 618 | 339 |

35

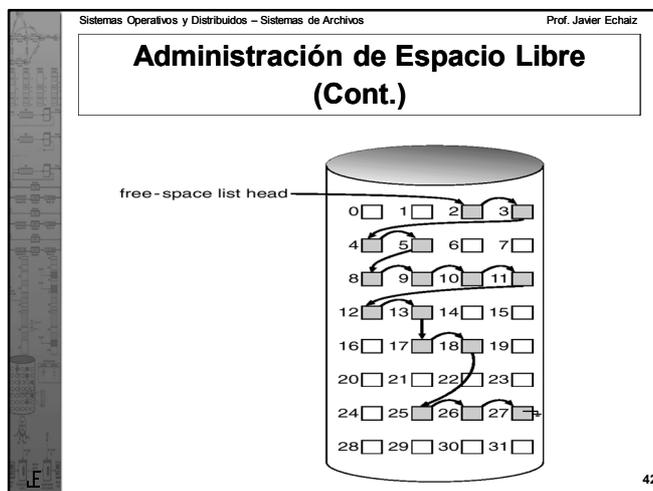
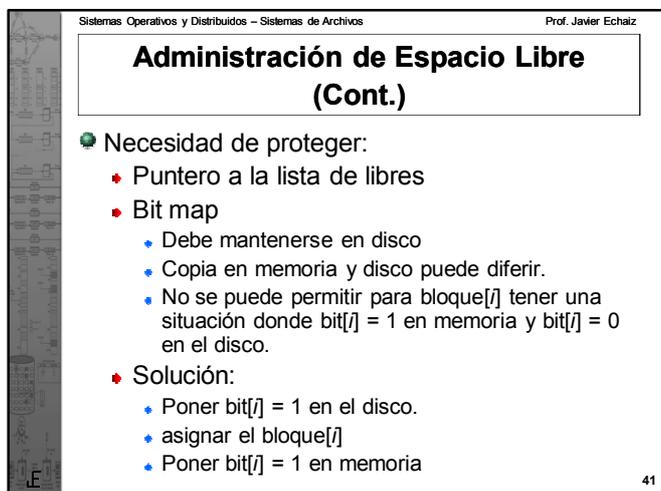
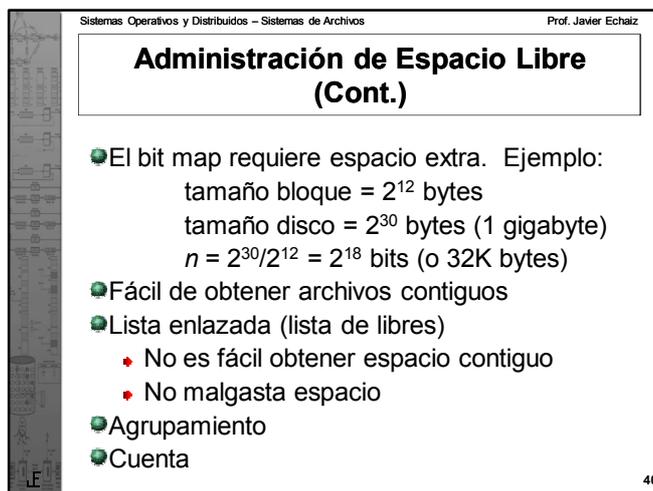
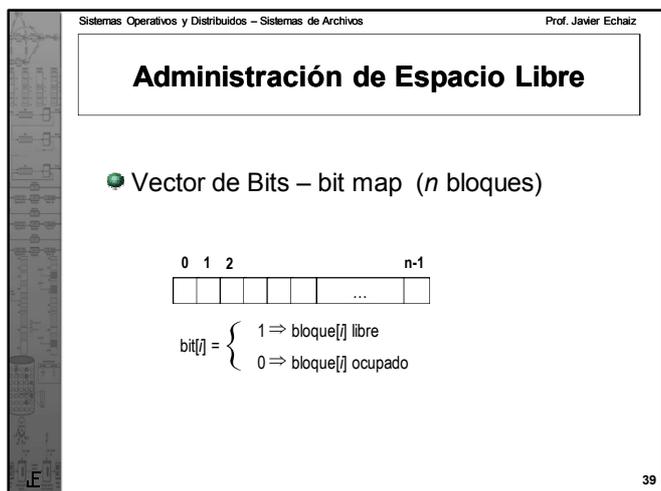
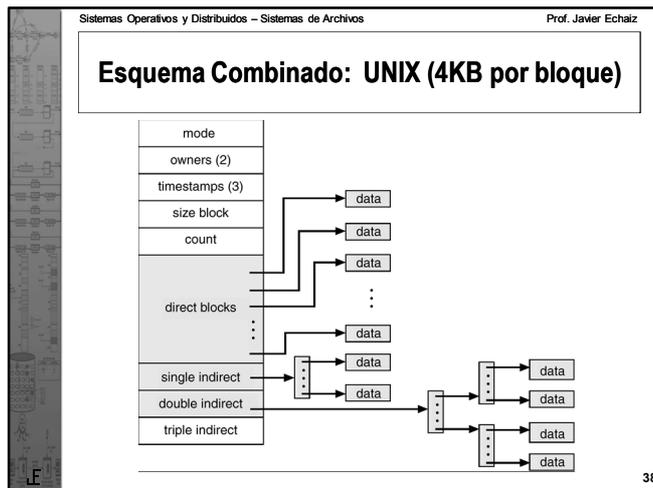
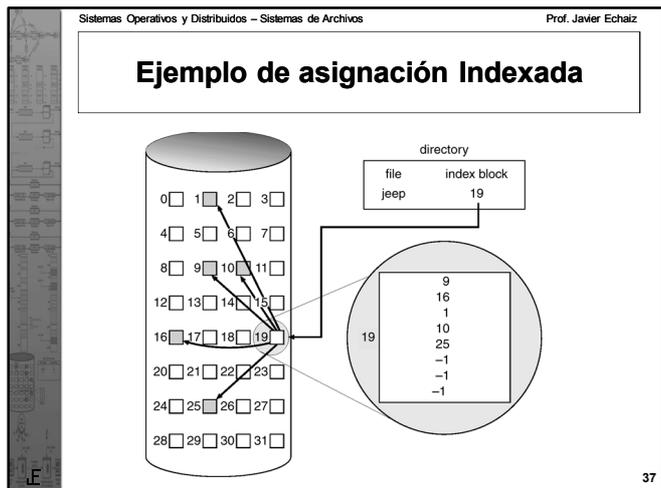
Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Asignación Indexada

- Pone todos los punteros juntos en bloque índice.
- Vista lógica.

Tabla de índices

36



Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Implementación de Directorio

- Lista lineal de nombres de archivos con punteros a los bloques de datos.
 - ♦ simple de programar
 - ♦ consume mucho tiempo en la ejecución
- Tabla *hash* – Lista lineal con estructura de datos *hash*.
 - ♦ Decrece el tiempo de búsqueda en el directorio
 - ♦ *colisiones* – situaciones donde dos nombres de archivos van a la misma ubicación
 - ♦ Tamaño fijo

43

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Eficiencia y Desempeño

- La eficiencia depende de:
 - ♦ asignacion en el disco y algoritmos de directorio
 - ♦ tipos de datos mantenidos en la entrada de directorio del archivos
- Desempeño
 - ♦ caché de disco – sección separada de memoria principal para bloques frecuentemente usados
 - ♦ *free-behind* y *read-ahead* – técnicas para optimizar el acceso secuencial
 - ♦ mejora del desempeño de la computadora dedicando una sección de la memoria como disco virtual, o disco RAM.

44

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Distintas Locaciones del Caché de Disco

The diagram illustrates the data flow between different components of a file system cache. On the left, the CPU is connected to the main memory. The main memory contains a 'ram disk' (a virtual disk in RAM), an 'open-file table', and a 'block buffer'. The main memory is connected to a 'controller', which is connected to a 'track buffer'. The track buffer is connected to the 'disk'.

45

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos Estructurado con Bitácora

- Un sistema de archivos estructurado con bitácora (log) registra cada actualización en el mismo como una transacción.
- Todas las transacciones son escritas en una bitácora. Una transacción se considera terminada (**committed**) una vez que es escrita en la bitácora. Sin embargo, el sistema de archivos puede no haber sido actualizado.
- Las transacciones en la bitácora son asincrónicamente escritas en el sistema de archivos. Cuando el sistema de archivos es modificado, la transacción es removida de la bitácora.
- Si el sistema de archivos cae, todas las transacciones remanentes en la bitácora pueden aún ser realizadas.

46

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura de Disco

- Los dispositivos de disco son vistos como un arreglo unidimensional de *bloques lógicos*, donde el bloque lógico es la más pequeña unidad de transferencia.
- Ese arreglo de bloques lógicos es mapeado secuencialmente en sectores del disco.
 - ♦ El sector 0 es el primer sector de la primera pista sobre el cilindro más externo.
 - ♦ El mapeo procede en orden a través de esa pista, luego el resto de las pistas en el cilindro, y luego el resto de los cilindros desde el más externo hasta el más interno.

47

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Planificación de Disco

- El sistema operativo es responsable de usar el hardware eficientemente — para los dispositivos de disco esto significa menor tiempo de acceso y mayor ancho de banda del disco.
- El tiempo de acceso tiene dos componentes importantes
 - *Tiempo de búsqueda*: es el tiempo que lleva mover las cabezas al cilindro que contiene el sector deseado.
 - *Latencia rotacional* es el tiempo adicional de espera por la rotación del disco hasta que el sector deseado está bajo las cabezas lectoras-escritoras.
- Minimizar el tiempo de búsqueda.
- Tiempo de búsqueda \approx distancia a la búsqueda
- El ancho de banda del disco es el número total de bytes transferidos, dividido por el total de tiempo entre el primer requerimiento de servicio y la terminación de la última transferencia.

48

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Planificación de Disco (Cont.)

- Existen varios algoritmos para planificar el servicio de los requerimientos de E/S.
- Se ilustran los mismos con una cola de requerimientos (0-199).

98, 183, 37, 122, 14, 124, 65, 67

La cabeza ha resuelto el requerimiento al sector 53

49

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Primero en Entrar- Primero en Salir FCFS

- Fácil de implementar
- Equitativo
- ¿Excesivas búsquedas?

Cola = 98, 183, 37, 122, 14, 124, 65, 67

Pistas: 640

50

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

El Tiempo de Búsqueda más Corto Primero SSTF

- Selecciona el requerimiento con el mínimo tiempo de búsqueda desde la posición que que ocupa la cabeza en ese momento.
- La planificación SSTF es una forma de planificación SJF; puede causar inanición de algunos requerimientos.
- Se muestra el mismo ejemplo anterior realizado con el algoritmo SSTF.

51

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

SSTF

- Minimiza tiempo de búsqueda
- El tiempo medio depende de la carga
- El tiempo de servicio es < cuando la cola es más larga!
- Puede llevar a esperas largas - inequitativo

Cola = 98, 183, 37, 122, 14, 124, 65, 67

FCFS: 640
Pistas: 236

52

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

SCAN

- El brazo del disco comienza en un extremo del disco y se mueva hacia el otro extremo, en su recorrido sirve todos los requerimientos hasta que llega al otro extremo donde se invierte el movimiento de la cabeza y continua sirviendo los requerimientos.
- Se lo llama, también algoritmo del ascensor.
- Se muestra el mismo ejemplo anterior implementando este algoritmo.

53

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

SCAN (Cont.)

Cola = 98, 183, 37, 122, 14, 124, 65, 67

FCFS: 640
SSTF: 236
Pistas: 236

54

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

LOOK

- Versión de SCAN
- El brazo va tan lejos en cada dirección como el último requerimiento.

Cola = 98, 183, 37, 122, 14, 124, 65, 67

FCFS: 640
SSTF: 236
Pistas: 208

55

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

C-SCAN (Circular SCAN)

- Provee un tiempo de espera más uniforme que el SCAN.
- La cabeza se mueve de un extremo a otro del disco sirviendo los requerimientos en el camino. Cuando alcanza el otro extremo inmediatamente retorna al comienzo del disco sin servir ningún requerimiento en ese viaje de retorno.
- Trata los cilindros como una lista circular que salta desde el último cilindro al primero o viceversa, según sea la convención.

56

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

C-SCAN

FCFS: 640
SSTF: 236
Pistas: 183

Cola = 98, 183, 37, 122, 14, 124, 65, 67

57

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

C-LOOK

- Versión del C-SCAN
- El brazo solo va tan lejos como el último requerimiento en cada dirección, luego invierte la dirección inmediatamente, sin retornar al extremo del disco sino hasta el último requerimiento en esa dirección.

58

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

C-LOOK (Cont.)

FCFS: 640
SSTF: 236
Pistas: 153

Cola = 98, 183, 37, 122, 14, 124, 65, 67

59

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Administración de Disco

- **Formato en bajo nivel, o formato físico** — Divide un disco en sectores que el controlador de disco puede leer y escribir.
- El uso de un disco es para contener archivos, el sistema operativo necesita registrar sus propias estructuras de datos en el disco.
 - **Partición** de un disco en uno o varios grupos de cilindros.
 - **Formato lógico** o "hacer un sistema de archivos"
- **Boot block** inicializa el sistema.
 - El **bootstrap** está almacenado en ROM.
 - Programa cargador **bootstrap**.
- Métodos para administrar los bloques malos.

60

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Disco MS-DOS

sector 0 boot block
sector 1 FAT
root directory
data blocks (subdirectories)

Veremos otros!

61

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Estructura RAID

- RAID – múltiples discos proveen **confiabilidad** via **redundancia**.
- RAID es establecido en seis niveles diferentes.

62

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Niveles RAID

(a) RAID 0: non-redundant striping
(b) RAID 1: mirrored disks
(c) RAID 2: memory-style error-correcting codes
(d) RAID 3: bit-interleaved Parity
(e) RAID 4: block-interleaved parity
(f) RAID 5: block-Interleaved distributed parity
(g) RAID 6: P + Q redundancy

63

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Niveles de RAID

| Categoría | Nivel | Descripción | Ritmo de E/S (Read/Write) | Ritmo Transfer de Datos (Read/Write) | Aplicaciones Típicas |
|----------------------|-------|---|---------------------------|--------------------------------------|--|
| Striping | 0 | No redundante | Large strips: Excelente | Pequeños strips: Excelente | Aplicaciones que requieren alta performance para datos no críticos |
| | 1 | Espejado | Buena/Medio | Medio/Medior | Drives de sistema, Archivos críticos |
| Acceso Paralelo | 2 | Redundante a través del código Hamming | Pobre | Excelente | |
| | 3 | Entrelazado de bit con paridad | Pobre | Excelente | Aplicaciones con grandes requerimientos de I/O, tales como imágenes, CAD |
| Acceso Independiente | 4 | Entrelazado de Bloques con paridad | Excelente/Medio | Medio/Pobre | |
| | 5 | Entrelazado de Bloques distribuido Paridad | Excelente/Medio | Medio/Pobre | Requerimientos con promedios altos, lecturas intensivas, búsqueda de datos |
| | 6 | Entrelazado de Bloques dual Paridad distribuida | Excelente/Pobre | Medio/Pobre | Aplicaciones con altos requerimientos de disponibilidad |

64

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Implementación de Almac. Estable

- Los esquemas de log (bitácora) de escritura adelantada requieren almacenamiento estable.
- Para implementar el almacenamiento estable:
 - Replicar información sobre más de un medio de almacenamiento no volátil con modo de fallas independientes.
 - Actualizar información de manera controlada para asegurar que se puede recuperar el dato estable luego de una falla durante la transferencia o recuperación.

65

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Dispositivos de almacenamientos Terciarios

- Bajo costo es la característica definida de los almacenamientos terciarios.
- Generalmente, el almacenamiento terciario es establecido sobre *medios removibles*.
- Ejemplos comunes son: dispositivos de memoria flash y CD-ROMs, DVDs, etc.

66

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Cintas

- Comparada al disco, una cinta es menos costosa y almacena más datos, pero el acceso aleatorio es mucho más lento.
- La cinta es un medio económico para propósitos que no requieren acceso aleatorio rápido, p.e., copias de backup de datos de disco, pues mantienen gigantescos volúmenes de datos.
- Las grandes instalaciones usan cambiadores robóticos de cintas que mueven cintas entre dispositivos y las almacena en estantes de una librería de cintas.
 - *stacker* – librería que mantiene varias cintas
 - *silos* – librería que mantiene miles de cintas

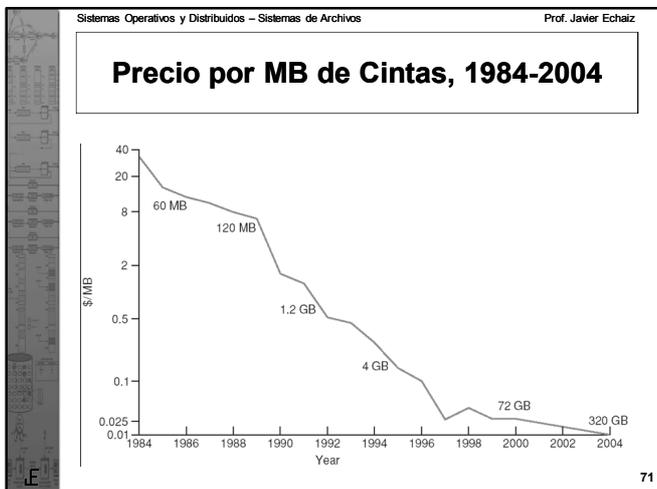
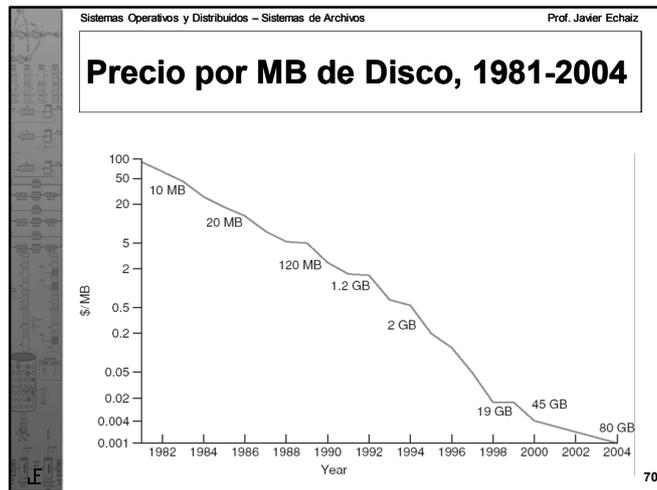
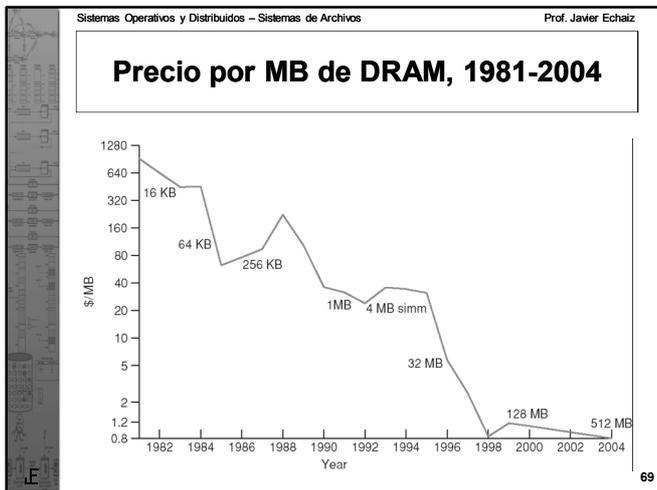
67

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Aspectos del Sistema Operativo

- Una de las mayores tareas es administrar los dispositivos físicos y presentar una abstracción de máquina virtual para las aplicaciones.
- Para los discos duros, el SO provee dos abstracciones:
 - Dispositivos crudos (raw) – un arreglo de bloques de datos.
 - Sistemas de archivos – el SO encola y planifica los requerimientos entrelazados de varias aplicaciones.

68



Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Extensión a Sistemas de Archivos Distribuidos

72

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Propósitos para el uso de archivos:

- almacenamiento de Información permanente
- Información compartida

También soporta:

- ⇒ Compartir información remota
- ⇒ Usuarios móviles
- ⇒ Disponibilidad (réplicas)
- ⇒ Estaciones de trabajo sin disco

73

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Servicios provistos por el sistema de archivos:

- Servicio de almacenamiento.
- Servicio de nombres.
- Servicio de archivos.

74

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Servicio de almacenamiento.

- ↳ asignación y manejo del espacio
- ↳ servicio de disco
- ↳ servicio de “bloqueo”

Servicio de nombres

- ↳ mapeo entre nombres externos e internos

75

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Servicio de archivos:

- ↳ acceso
- ↳ semántica de archivos compartidos
- ↳ caching
- ↳ replicación
- ↳ control de concurrencia

76

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Características Deseables de los SAD

Transparencia

- ↳ de estructura: no se conoce el número de servidores ni sus lugares
- ↳ de acceso
- ↳ de nombres
- ↳ de replicación

77

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Movilidad de usuario: el usuario no está obligado a trabajar en un nodo único.

Rendimiento: se mide como la cantidad de tiempo que demora en satisfacer el requerimiento de un cliente.

Simplicidad y facilidad de uso: semántica fácil de entender.

78

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Escalabilidad: se debe adaptar al crecimiento de nodos y usuarios en el sistema.

Alta disponibilidad: atento a fallas.

Alta confiabilidad: almacenamiento estable.

Integridad de datos: control de concurrencia (transacciones atómicas)

79

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Seguridad: debe ser lo suficientemente seguro como para que los usuarios puedan confiar en la privacidad de sus datos.

Heterogeneidad: se torna inevitable como consecuencia de la gran variedad de equipamiento y software.

80

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Modelos de Archivos

Los archivos pueden ser:

- ◀estructurados: son raros hoy, son una secuencia de registros (indexados o no indexados)
- ◀no estructurados: UNIX.
- ◀mutables
- ◀inmutables

81

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Modelos de Acceso a Archivos

- ▣ Modelo de servicio remoto.
- ▣ Modelo de captura de datos (caching): trae consigo problemas de consistencia.

82

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Unidades de transferencia de datos:

- ◀Modelo de transferencia a nivel de archivos.
- ◀Modelo de transferencia bloques.
- ◀Modelo de transferencia bytes.
- ◀Modelo de transferencia por registros.

83

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Semántica de Archivos Compartidos

- ▣ Semántica de sesión
- ▣ Semántica de archivos inmutables
- ▣ Semántica de transacciones

84

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Esquemas de caché

En centralizados:

- granularidad
- tamaño
- políticas de reemplazo

En distribuidos se agrega:

- ubicación
- propagación de la modificación
- validación

85

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Ubicación del caché

- en memoria del servidor
- en disco del cliente
- en memoria del cliente

86

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

| Ubicación caché | Costo de acceso con éxito en el caché. | Ventajas |
|----------------------|--|---|
| Memoria del servidor | Un acceso por la red | <ul style="list-style-type: none"> •Fácil de implementar •Totalmente transparente a los clientes •Fácil de mantener consistente el archivo original y los datos en el caché •Fácil para soportar semántica UNIX |
| Disco del cliente | Un acceso a disco | <ul style="list-style-type: none"> •Confiabilidad en caso de "crash" •Gran capacidad de almacenamiento •Adecuado para soportar operación sin conexión •Contribuye a la escalabilidad y confiabilidad |
| Memoria del cliente | — | <ul style="list-style-type: none"> •Máxima ganancia de rendimiento •Permite estaciones de trabajo sin disco •Contribuye a la escalabilidad y confiabilidad |

87

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

1. Sin caché
2. Caché localizado en la memoria del servidor
3. Caché localizado en el disco del cliente
4. Caché localizado en la memoria del cliente

ubicación original del archivo

88

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Propagación de la Modificación

- write through
- delayed write
 - ⇒ escritura cuando se "echa" la información del caché
 - ⇒ escritura periódica
 - ⇒ escritura cuando se cierra

89

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Validación de cachés

- iniciado por el cliente
 - ↳verificación antes de cada acceso.
 - ↳verificación periódica.
 - ↳Verificación sólo cuando el archivo es abierto para el uso.
- iniciado por el servidor

90

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Replicación

Ventajas:

- <se incrementa la disponibilidad
- <se incrementa la confiabilidad
- <mejora el tiempo de respuesta
- <reduce el tráfico en la red
- <mejora el procesamiento total
- <buena escalabilidad
- <operación autónoma

91

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Transparencia de replicación

- Replicación explícita
- Replicación implícita/relajada

Problema de actualización de múltiples copias

Está relacionado con mantener consistentes las copias

92

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Replicación read-only: generalmente código

Protocolo read-any/write-all: no puede manejar las redes partidas

Protocolo de copias disponibles

Protocolo de copias primarias

93

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Protocolos basados en quorum: de un total de n copias del archivo F , un número de r deben ser leídas (quorum de lecturas), de la misma forma w copias para escritura (quorum de escritura) tal que:

$$(r + w) > n$$

$$w > n/2$$

94

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Protocolos derivados:

- <leer alguno y escribir todos $r = 1$ $w = n$
- <leer todos y escribir alguno $r = n$ $w = 1$
- <consenso de mayoría

95

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Protocolos Basados en Quorum

Quorum lectura

Quorum escritura

(a) $N_R = 3, N_W = 10$

(b) $N_R = 7, N_W = 6$

(c) $N_R = 1, N_W = 12$

Tres ejemplos del algoritmo de votación:

- a) Una correcta elección de conjuntos de lectura y escritura.
- b) Una elección que puede llevar a conflictos escritura-escritura.
- c) Una elección correcta, conocida como ROWA (read one, write all)

96

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Tolerancia a Fallas

Hay propiedades que influncian directamente en un SAD para que sea tolerante a las fallas:

- <Disponibilidad ⇒ replicación
- <Robustez
- <Almacenamiento estable:
 - . volátil
 - . discos
 - . estable

97

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Efecto del paradigma de servicio en la tolerancia a las fallas

- Servidores con estado
- Servidores sin estado

98

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Proceso Cliente

```

Open(filename,mode)
Return(fid)
Read(fid,100,buf)
Return(bytes 0 a 99)
Read(fid,100,buf)
Return(bytes 100 a 200)
                    
```

Proceso Servidor

Tabla de archivos

| fid | Modo | Puntero R/W |
|-----|------|-------------|
| | | |
| | | |
| | | |

Dos subsecuentes lecturas de 100 bytes (200 bytes en total)

99

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Proceso Cliente

Información de estado de archivos

| Nombre de archivo | Modo | Puntero R/W |
|-------------------|------|-------------|
| | | |
| | | |
| | | |

Proceso Servidor

```

Read(filename,0,100,buf)
Return(bytes 0 a 99)
Read(filename,100,100,buf)
Return(bytes 100 a 199)
                    
```

Dos subsecuentes lecturas de 100 bytes (200 bytes en total)

100

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Diferencias entre Servicios con estado y sin estados

Recuperación de Fallas

- Un servidor con estados pierde, en un *crash*, todo su estado volátil.
 - Restaura el estado por un protocolo basado en un diálogo con clientes o aborta las operaciones que se estaban llevando a cabo cuando el *crash* ocurrió.
 - El servidor necesita estar al tanto de las fallas en los clientes para reclamar el espacio reservado para registrar el estado de los procesos de los clientes caídos

101

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

- Con un servidor sin estados, los efectos de fallas y recuperación en el servidor no son notables. Un nuevo servidor *encarnado* puede responder sin dificultad a un requerimiento autocontenido.

Penalización por usar un servicio robusto sin estados:

- Mensajes de requerimiento más grandes
- Menor procesamiento de requerimientos
- Restricciones adicionales al diseño de un SAD

102

Sistemas Operativos y Distribuidos – Sistemas de Archivos Prof. Javier Echaiz

Sistema de Archivos en Sistemas Distribuidos

Algunos ambientes requieren servicio con estado

- Un servidor que emplea validación de caché iniciada por el servidor, dado que mantiene un registro de todos los archivos que están *cached* por varios clientes.
- El uso de descriptores de archivos en UNIX y los *offset* implícitos es inherente con estado, los servidores deben mantener las tablas para mapear los descriptores de archivos a los nodos y almacenar los *offsets* corrientes en un archivo.

103

