



Sistemas Operativos

El sistema de Entrada/Salida

Diego Madariaga

Algunos conceptos

- ▷ **Proceso intensivo en CPU:** requiere principalmente del poder del procesador para su ejecución, sin usar mayormente ningún otro componente (e.g. calcular series numéricas).
- ▷ **Proceso intensivo en E/S:** consiste principalmente en leer y escribir en un disco (e.g. copiar una gran cantidad de archivos)

Ejecución de procesos

Suponga que en un instante dado necesita ejecutar 4 procesos independientes entre sí en una máquina quad-core. Discuta si lanzaría los procesos simultáneamente, secuencialmente o en algún orden específico considerando que:

a) Todos los procesos son intensivos en CPU, pero no en E/S

Sol: Una máquina quad-core tiene cuatro núcleos independientes, lo cual permitirá ejecutar hasta cuatro procesos intensivos en CPU en forma paralela sin interferir entre ellos

Ejecución de procesos

Suponga que en un instante dado necesita ejecutar 4 procesos independientes entre sí en una máquina quad-core. Discuta si lanzaría los procesos simultáneamente, secuencialmente o en algún orden específico considerando que:

b) Todos los procesos son intensivos en E/S, pero no en CPU. Hay un solo disco

Sol: Si tenemos procesos intensivos en E/S sobre un solo disco, al ejecutarlos en paralelo lograremos que el disco (y en particular el cabezal) sea un cuello de botella en la ejecución. El tiempo de ejecución podría ser mayor a que hayan sido lanzados secuencialmente (donde cada uno usaría el disco en forma exclusiva)

Ejecución de procesos

Suponga que en un instante dado necesita ejecutar 4 procesos independientes entre sí en una máquina quad-core. Discuta si lanzaría los procesos simultáneamente, secuencialmente o en algún orden específico considerando que:

c) 2 procesos son solo intensivos en E/S y los otros 2 procesos son solo intensivos en CPU. Hay un solo disco.

Sol: Si tenemos dos procesos intensivos en CPU (P1 y P2) y dos procesos intensivos en E/S sobre un solo disco (P3 y P4) podríamos lanzar en forma paralela P1, P2 y P3 y una vez finalizado P3 iniciar P4 a continuación.

Inodes y bloques de datos

inodo contiene:

- ▷ 12 punteros a bloques de datos,
- ▷ 1 puntero a un bloque de indirección simple,
- ▷ 1 puntero de indirección doble y
- ▷ 1 puntero de indirección triple.

Un bloque de indirección simple es un bloque de datos que en vez de contener bytes del archivo, contiene punteros a otros bloques de datos que contendrán bytes del archivo. Cada puntero a bloque de datos es de 32 bits.

Inodes y bloques de datos (1)

En un sistema Unix un archivo tiene un tamaño de $(12+256)*512+1$ bytes.

Explique cuántos bloques de datos y bloques de indirección se utilizarán, suponiendo que el tamaño de los bloques de la partición es **512** bytes.

Inodes y bloques de datos (1)

Tamaño de archivo: $(12+256)*512+1$ bytes.

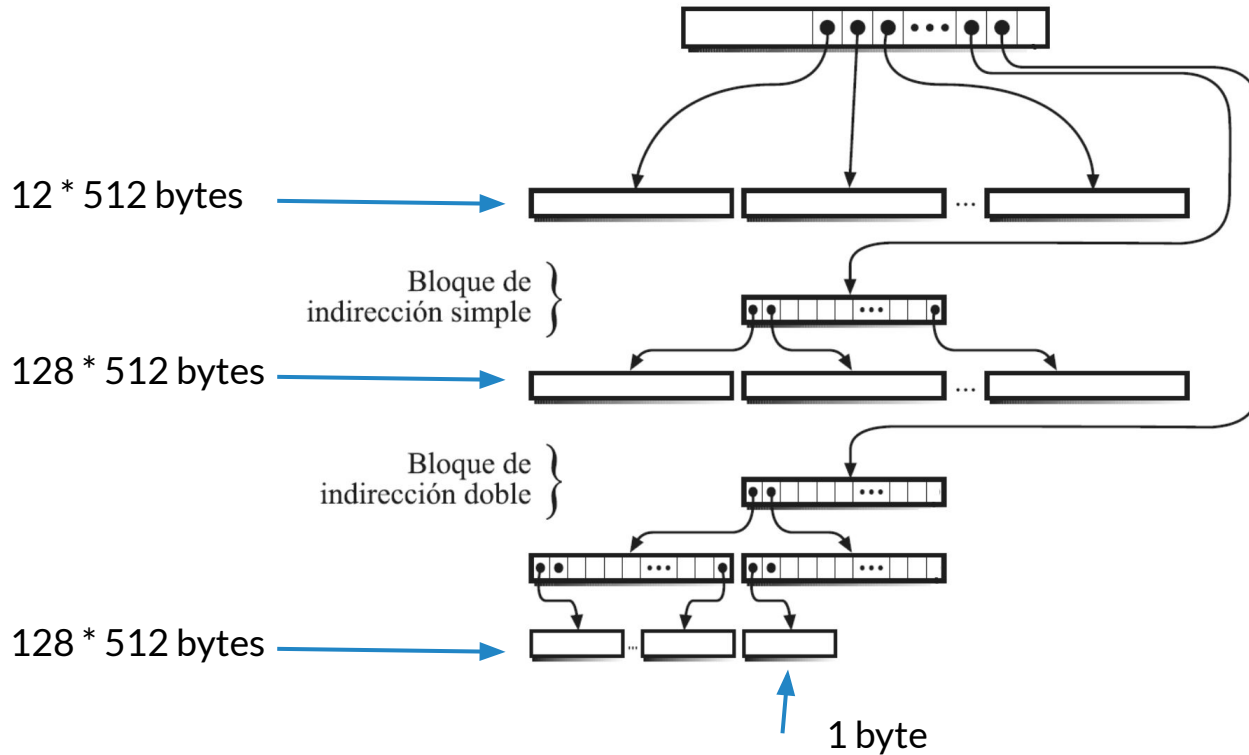
Si el tamaño de bloque es de 512 bytes, entonces cada bloque de indirección podrá contener hasta 128 punteros (4 bytes * 128).

Utilizando los 12 punteros de datos que hay en el inodo podremos referenciar $12 * 512$ bytes.

Con el puntero de indirección simple podremos referenciar $128 * 512$ bytes.

Con el puntero de indirección doble podremos referenciar un máximo de $128 * 128 * 512$ bytes, lo que es suficiente para contener los $128 * 512 + 1$ bytes faltantes (deberemos ocupar y referenciar 129 bloques de datos)

Inodes y bloques de datos (1)



Inodes y bloques de datos (1)

Suponga que un programa usa lseek para posicionarse en el último byte del archivo y lo lee. Explique cuántos bloques será necesario leer. Suponga que el caché de disco está completamente vacío.

R. Debemos leer 4 bloques: el bloque del inodo, el bloque de indirección doble en primer y segundo nivel y el bloque de datos.

Inodes y bloques de datos (2)

En un sistema Unix un archivo tiene un tamaño de $(12+256)*512+1$ bytes.

Explique cuántos bloques de datos y bloques de indirección se utilizarán, suponiendo que el tamaño de los bloques de la partición es **4096** bytes.

Inodes y bloques de datos (2)

Tamaño de archivo: $(12+256)*512+1$ bytes = 134 kB + 1 byte

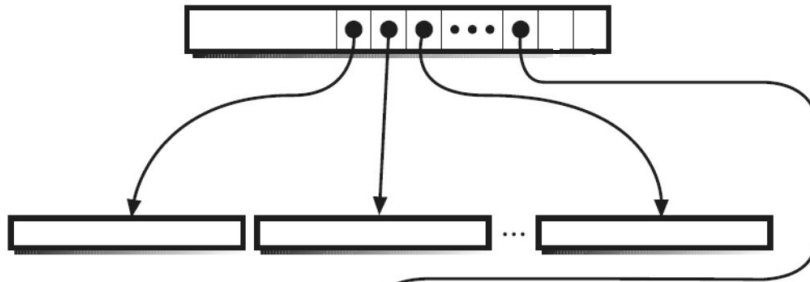
Si el tamaño de bloque es de 4096 bytes, entonces cada bloque de indirección podrá contener hasta 1024 punteros (4 bytes * 1024).

Utilizando los 12 punteros de datos que hay en el inodo podremos referenciar $12 * 4096$ bytes = 48 kB

Con el puntero de indirección simple podremos referenciar $1024 * 4096$ bytes = 4096 kB, lo que es suficiente para contener los 86 kB + 1 byte restantes (deberemos ocupar y referenciar 22 bloques de datos)

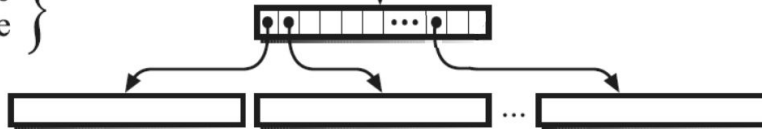
Inodes y bloques de datos (2)

12 * 4096 bytes
= 48 kB



Bloque de
indirección simple }

22 bloques de 4 kB
= 88 kB



(suficiente para los 86kB + 1 byte restantes)

Inodes y bloques de datos (2)

Suponga que un programa usa lseek para posicionarse en el último byte del archivo y lo lee. Explique cuántos bloques será necesario leer. Suponga que el caché de disco está completamente vacío.

R. Debemos leer 3 bloques: el bloque del inodo, el bloque de indirección simple y el bloque de datos.

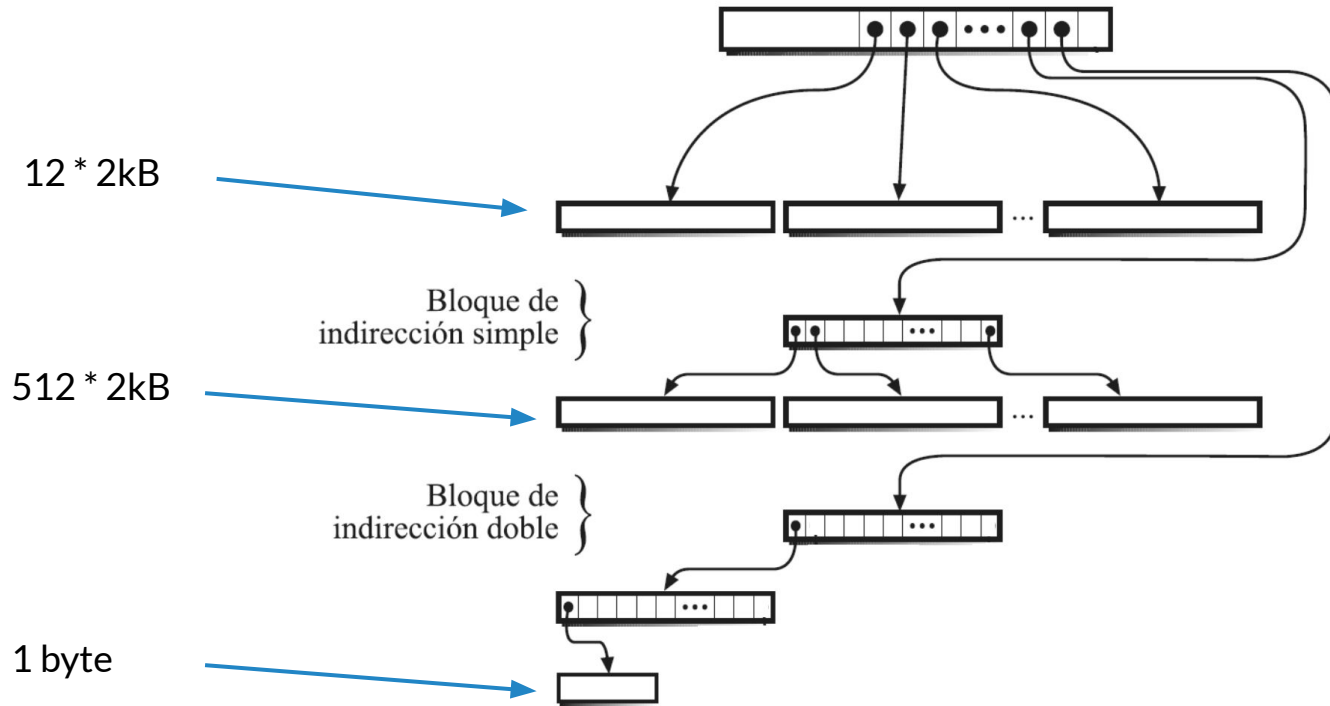
Inodes y bloques de datos (3)

Se tiene un archivo que no requiere bloques de indirección doble en una partición Unix con bloques de 2 kB. Se agrega un byte a este archivo y se crea el bloque de indirección doble. Haga un diagrama mostrando inodo, bloques de datos y de indirección. ¿De qué tamaño es el archivo?

R. Dado que el bloque de indirección es de 2 kB, contiene 512 punteros a bloques de datos (4 bytes * 512).

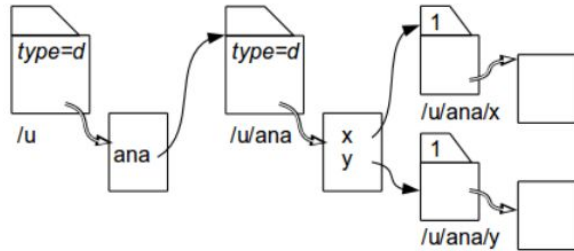
Por lo tanto el archivo es de $12 * 2\text{kB} + 512 * 2\text{kB} + 1 \text{ byte} = 1048\text{kB} + 1$

Inodes y bloques de datos (3)



Archivos y directorios

La figura muestra varios archivos y directorios de la partición /u en un sistema Unix:

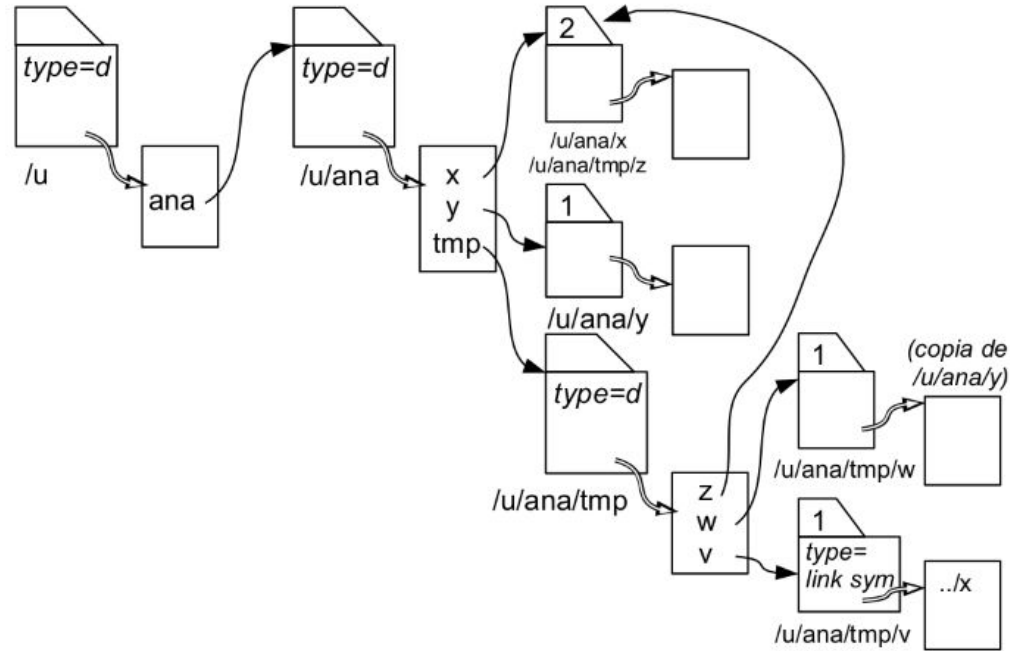


La usuaria ana ejecuta los siguientes comandos:

```
% mkdir /u/ana/tmp  
% cd /u/ana  
% ln x tmp/z  
% cp y tmp/w  
% cd tmp  
% ln -s ../x v
```

Rehaga la figura de acuerdo a los cambios realizados

Archivos y directorios



Disco moderno vs SSD

Suponga que en un sistema existen solo 2 procesos en ejecución. Ambos leen secuencialmente archivos de gran tamaño en la misma partición con bloques de 1kB.

El primero de los archivos se encuentra localizado en las pistas externas del disco, mientras que el segundo en las pistas internas.

Ambos procesos leen los archivos en trozos de 1kB usando read. Estime la velocidad de lectura de estos archivos considerando los siguientes tipos de almacenamiento secundario:

Disco moderno vs SSD

a) Un disco moderno con tiempo de acceso y velocidad de transferencia razonables, el núcleo no implementa read-ahead.

R. Supongo disco de 10 milisegundos de tiempo de acceso y tasa de transferencia de 100 MB/seg. Cada bloque requiere desplazar el cabezal del disco y por lo tanto cuesta 10 milisegundos. El tiempo de transferencia es marginal. La velocidad total de lectura es de 100 kB/seg. Para cada archivo es de 50 kB/seg. (Se pueden considerar razonables tiempos de acceso entre 3 a 30 milisegundos y tasas de transferencia entre 10 y 200 MB/seg.)

Disco moderno vs SSD

b) El mismo disco moderno, el núcleo implementa read-ahead de 100 bloques.

R. Cada lectura lee 100 KB y requiere 10 milisegundos para desplazar el cabezal y 0.1 milisegundos de tiempo de transferencia (marginal). La velocidad total de lectura es de 10 MB/seg o 5 MB/seg para cada archivo.

Disco moderno vs SSD

c) Un SSD (solid state drive), el núcleo no implementa read-ahead.

R. Considero una tasa de transferencia de 200 MB para el SSD. Como no hay que mover el cabezal, en teoría la velocidad total de lectura es de 200 MB/seg. (En la práctica se alcanzan velocidades de unos 40 MB/seg. Tasas de transferencia razonables para un SSD son de 50 a 2000 MB/seg.)



Sistemas Operativos

El sistema de Entrada/Salida

Diego Madariaga