
Auxiliar 12:

El Sistema de Entrada/Salida

— José Astorga —

[PPT adaptada de las de Diego Madariaga](#)

1. Entrada/Salida

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 archivo)

Problema 1: 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 :

Problema 1: 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

R: 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.

Problema 1: 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:

Problema 1: 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

R: 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).

Problema 1: 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:

Problema 1: 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

R: 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.

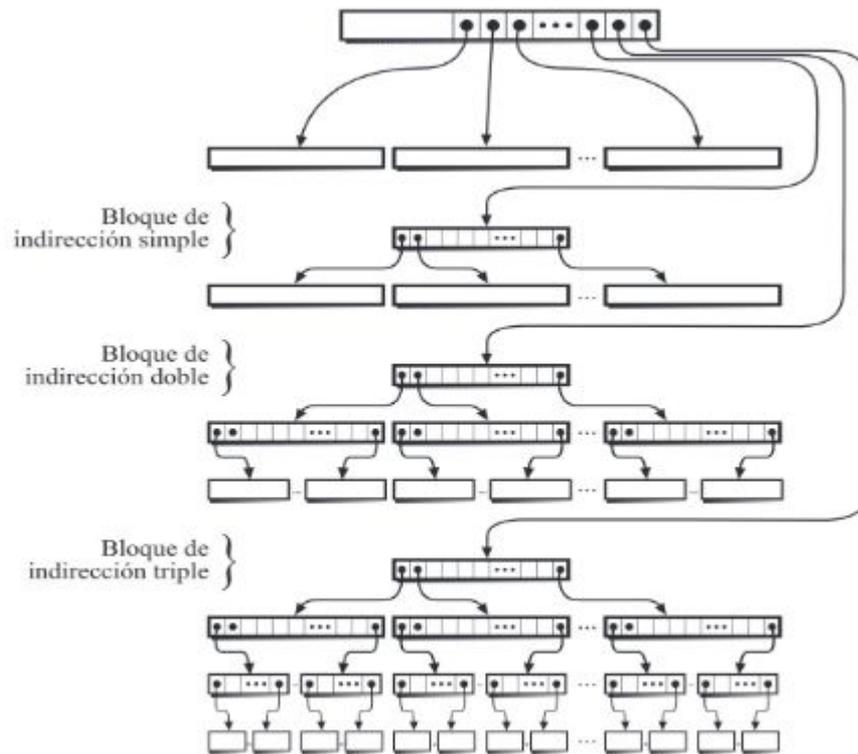
Inodos y Bloques de datos

Inodos y bloques de datos

Cada 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.
 - Un bloque de indirección doble contiene punteros a bloques de indirección simple.
 - Un bloque de indirección triple contiene punteros a bloques de indirección doble.
 - Consideramos punteros de 32 bits.

Diagrama de bloques de un archivo



Problema 2: Inodes y bloques de datos

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

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

Haremos un diagrama

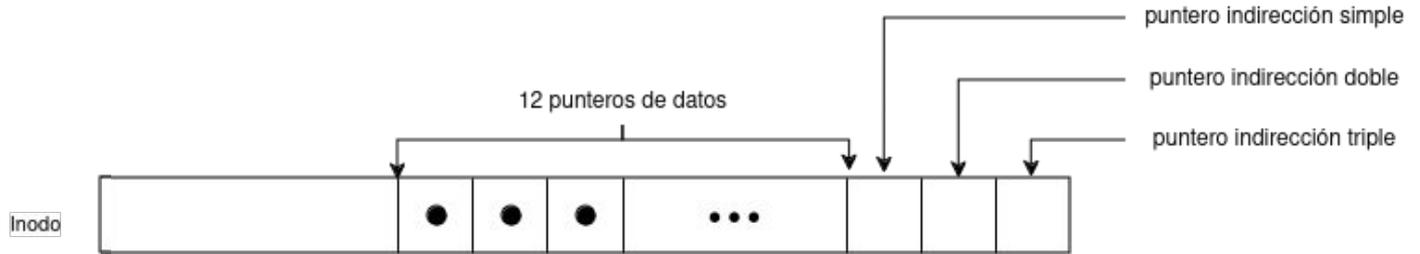
Problema 2: Inodes y bloques de datos

- Tamaño de archivo: **$(12 + 256) * 512 + 1$ Bytes.**
- Tamaño del bloque: **512B.**
- ¿Cuántos bloques de datos necesitaremos?
 - $\lceil \text{tamaño(archivo)} / \text{tamaño(bloque)} \rceil =$
 - $\lceil ((12 + 256) * 512 + 1) / 512 \rceil = \mathbf{12 + 256 + 1 = 269}$
- ¿Cuántos punteros caben en cada bloque de indirección?
 - Si el tamaño de bloque es de 512 bytes, entonces cada bloque de indirección podrá contener hasta **128 punteros (512B / 4B).**

Problema 2: Inodes y bloques de datos

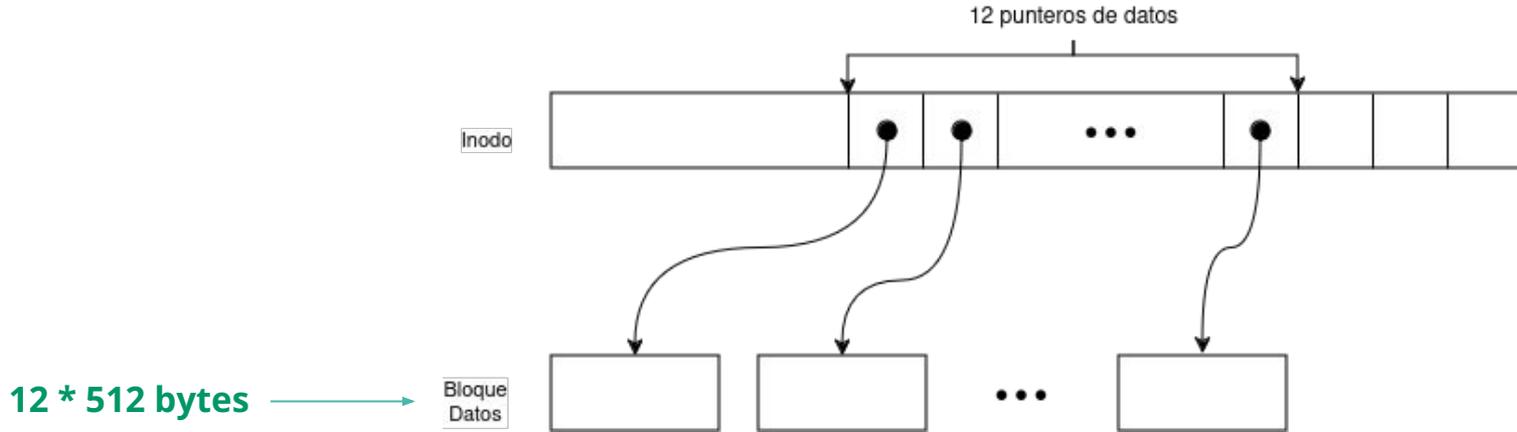
- ¿Cuanta memoria podemos direccionar en cada capa?
 - El inodo tiene 12 punteros a bloques, puede direccionar hasta:
 - **$12 * 512B$.**
 - Cada bloque de indirección simple podrá direccionar hasta:
 - **$\#punteros * tamaño_bloque = 128 * 512B$.**
 - Cada bloque de indirección doble podrá direccionar hasta:
 - **$\#punteros * tamaño\ direccionable\ por\ indirección\ simple = 128 * 128 * 512B$.**
 - Cada bloque de indirección triple podrá direccionar hasta:
 - **$\#punteros * tamaño\ direccionable\ por\ indirección\ doble = 128 * (128^2 * 512B)$**
 - **$(128^3 * 512B)$**

Problema 2: Inodes y bloques de datos



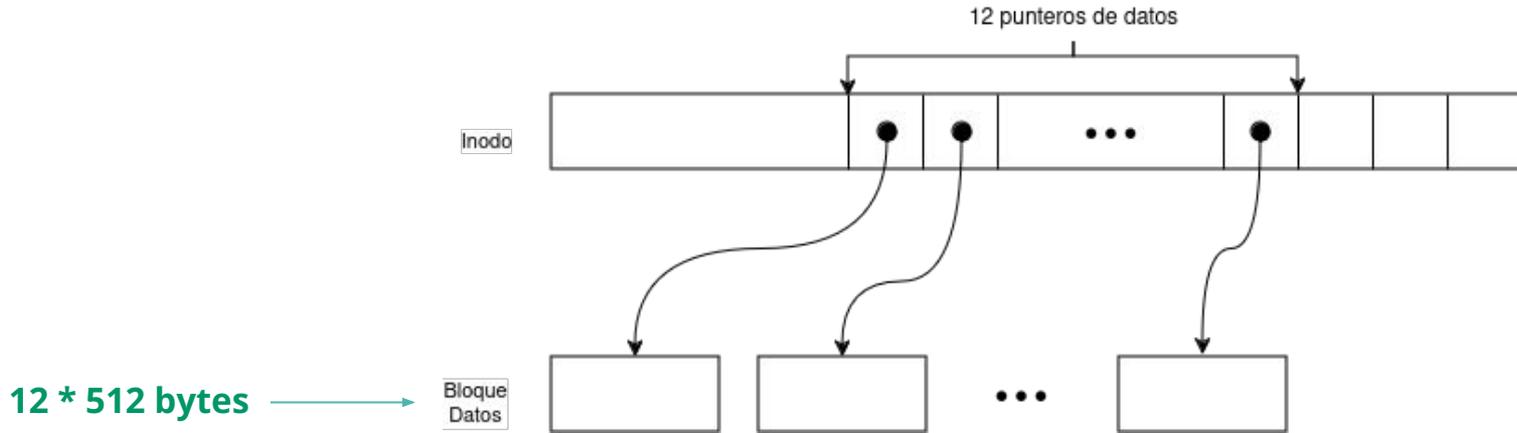
Necesitamos 269
bloques de datos.

Problema 2: Inodes y bloques de datos



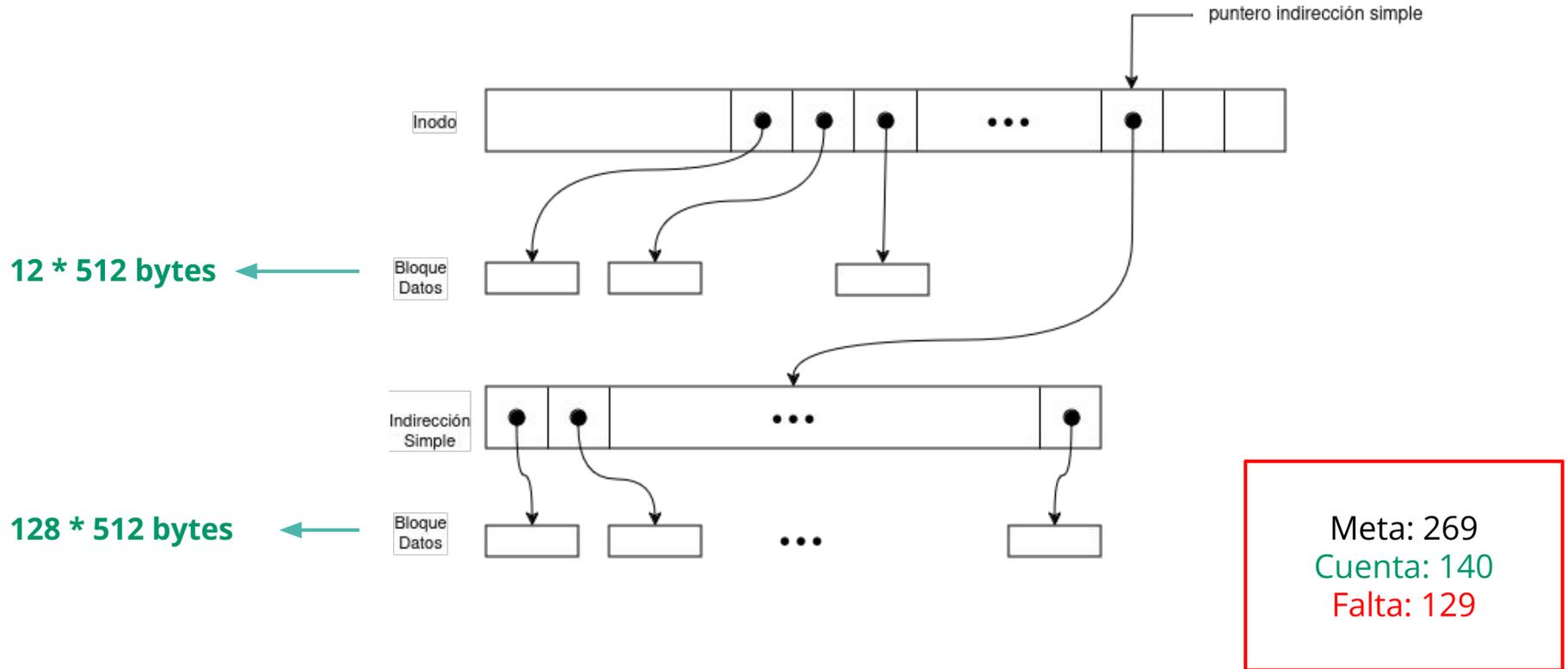
Necesitamos 269 bloques de datos.

Problema 2: Inodes y bloques de datos

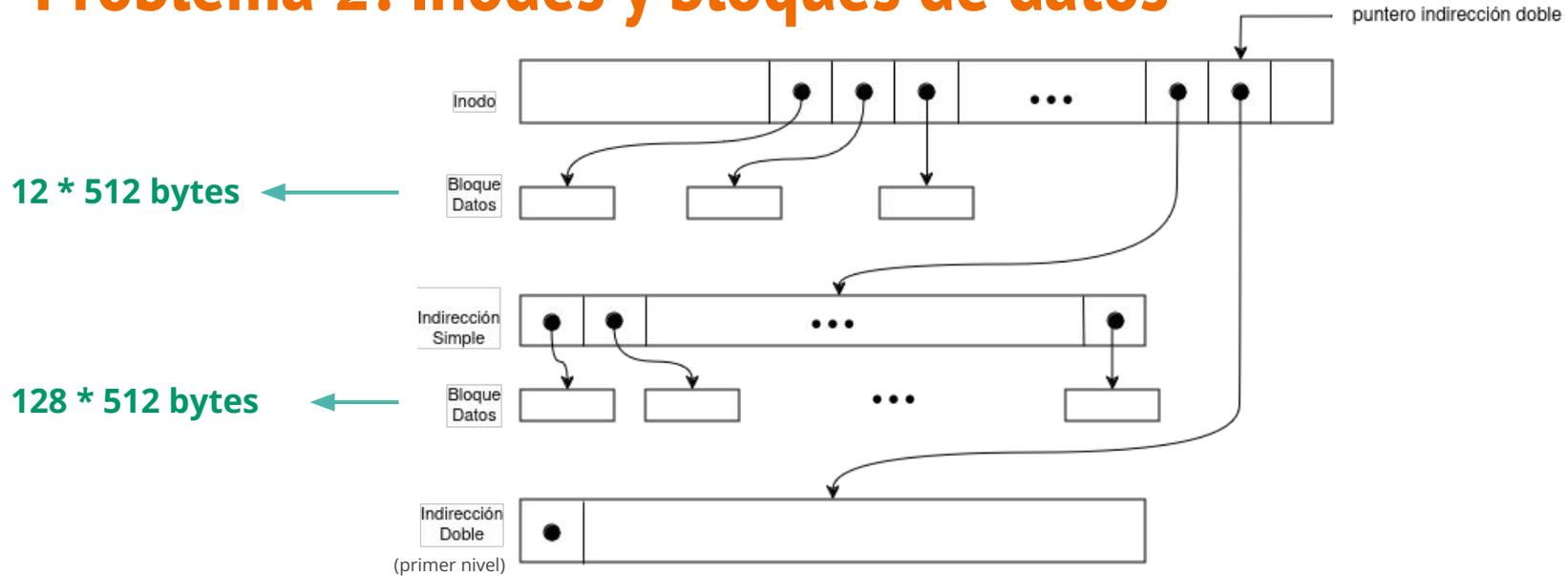


Meta: 269
Cuenta: 12
Falta: 257

Problema 2: Inodes y bloques de datos

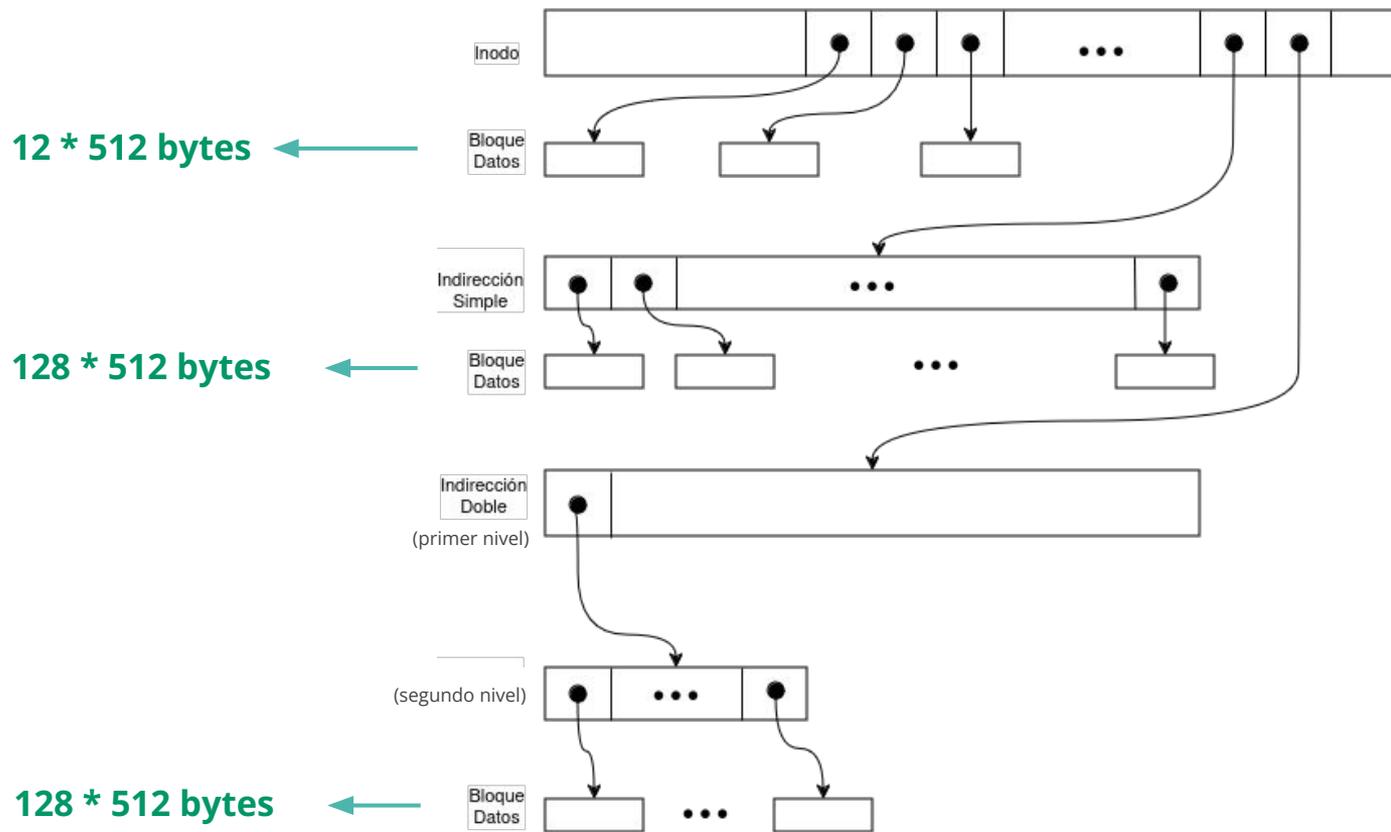


Problema 2: Inodes y bloques de datos



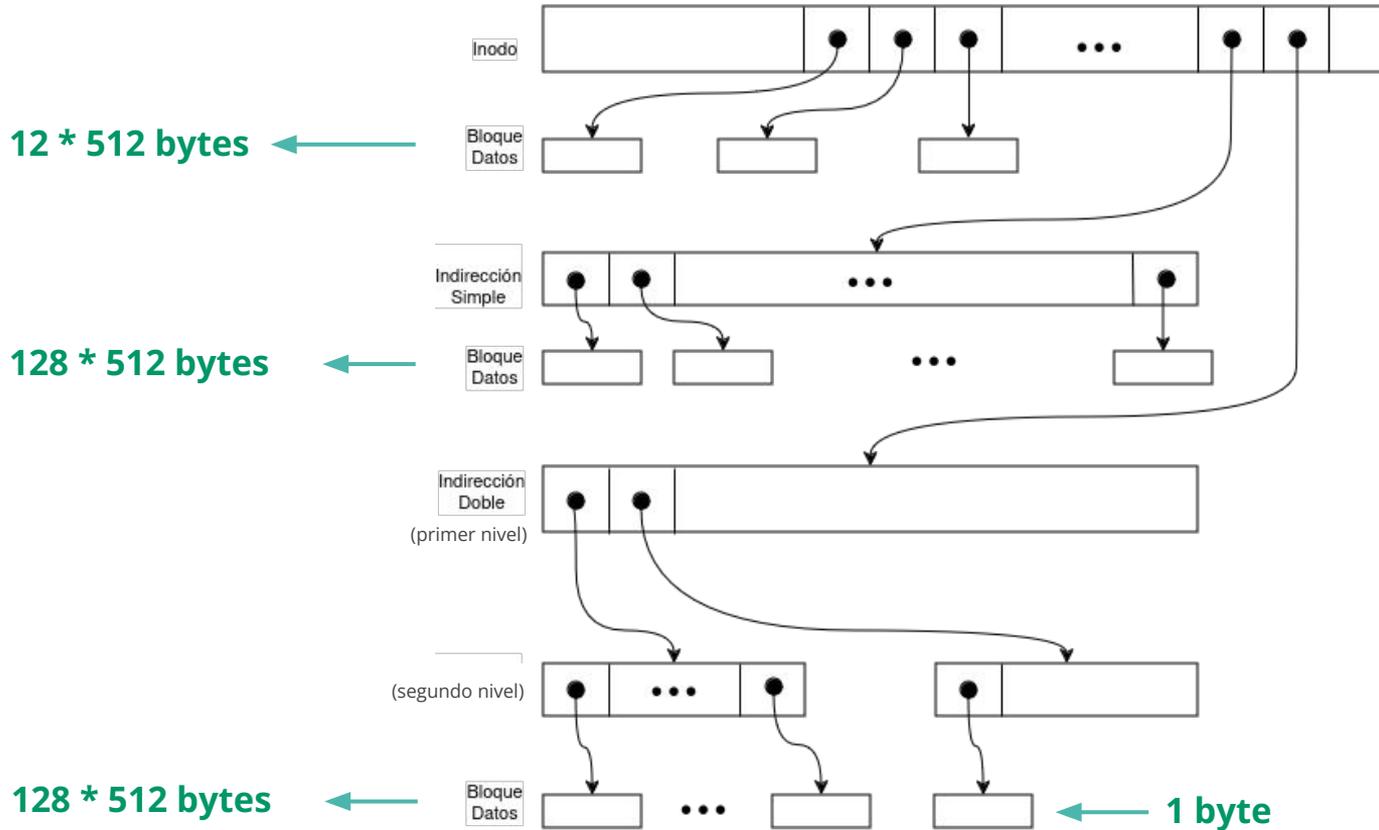
Meta: 269
Cuenta: 140
Falta: 129

Problema 2: Inodes y bloques de datos



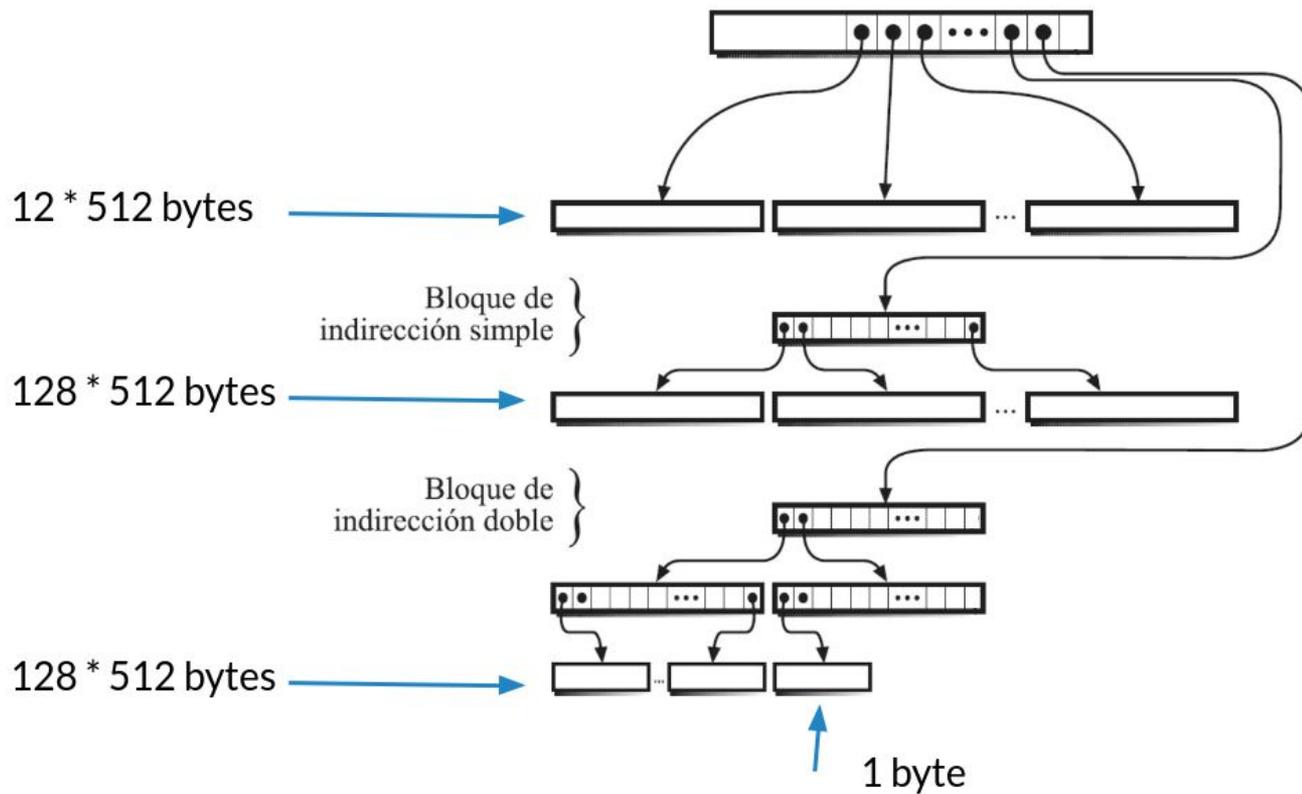
Meta: 269
Cuenta: 268
Falta: 1

Problema 2: Inodes y bloques de datos



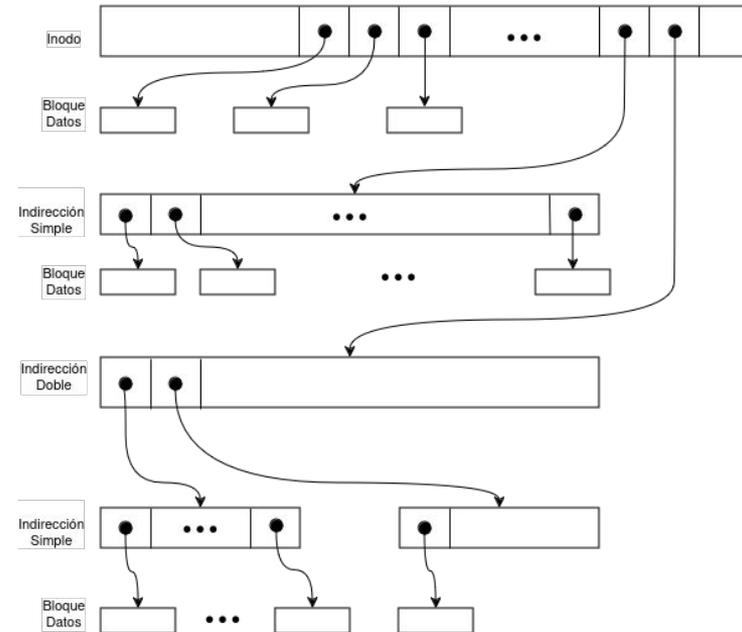
Meta: 269
Cuenta: 269
Falta: 0

Problema 2: Inodes y bloques de datos



Problema 2: Inodes y bloques de datos

b) 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.

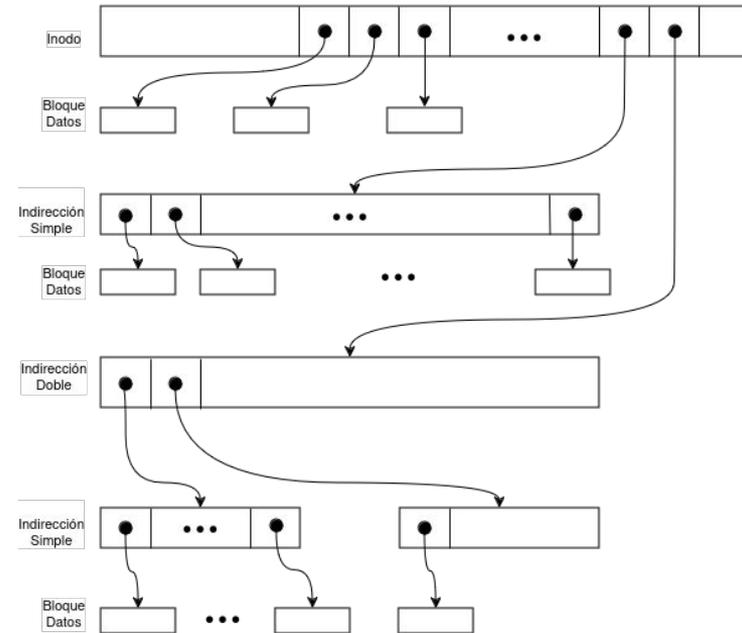


Problema 2: Inodes y bloques de datos

b) 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.

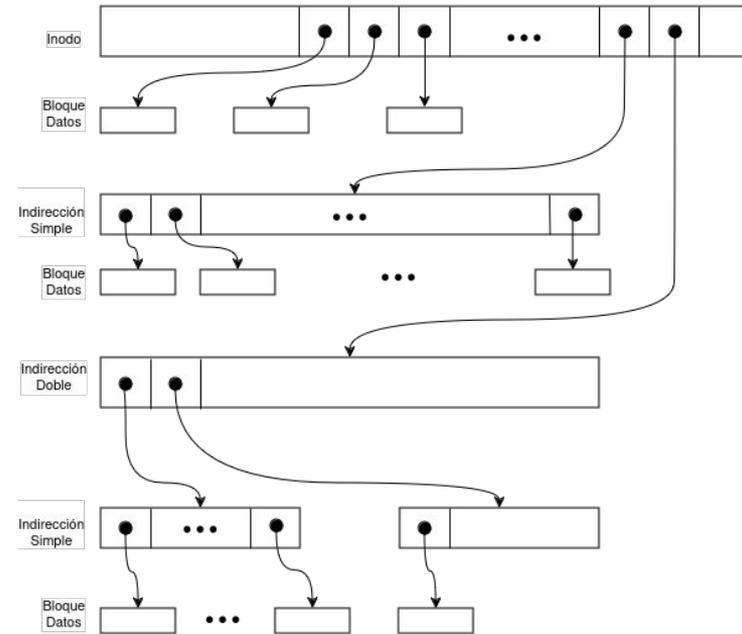
R. Debemos leer 4 bloques:

- El bloque que contenga al inodo
- El bloque de indirección doble en primer nivel
- El bloque de indirección doble en segundo nivel
- El bloque de datos.



Problema 2: Inodes y bloques de datos

c) ¿Cuánto espacio en disco se ocupa realmente para almacenar este archivo?



Problema 2: Inodes y bloques de datos

c) ¿Cuánto espacio en disco se ocupa realmente para almacenar este archivo?

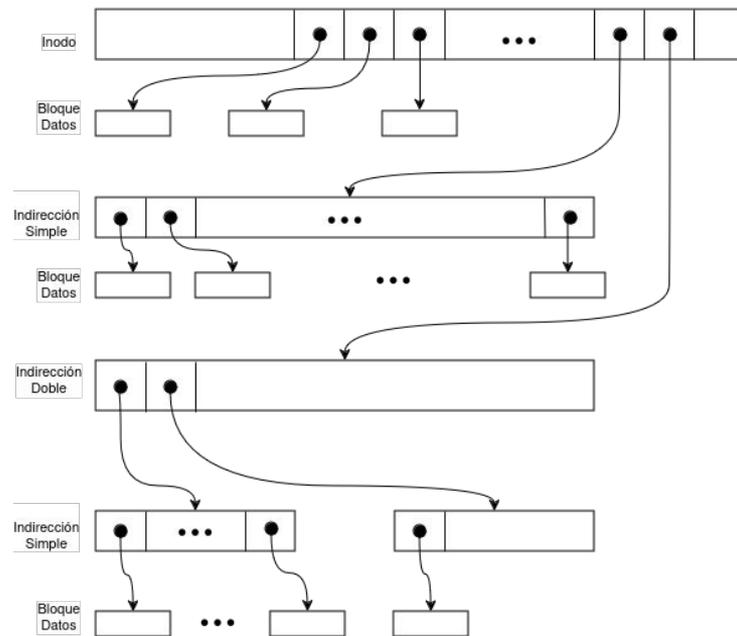
R. Debemos contar todos los bloques que utiliza el archivo, tanto de datos como de indirección:

- Se ocupa un total de 269 bloques de datos (12 + 128 + (128 + 1)).
- 4 bloques de indirección (1 simple + 1 doble con 2 simples).

Tamaño total = #bloques * tamaño(bloque) = 273 * 512 bytes = **136.5kB**

También podemos considerar el tamaño del inodo (depende del sistema, generalmente 128B).

En total son 136.5kB + inodo.



Problema 3: Inodes y bloques de datos

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

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

Problema 3: Inodes y bloques de datos

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

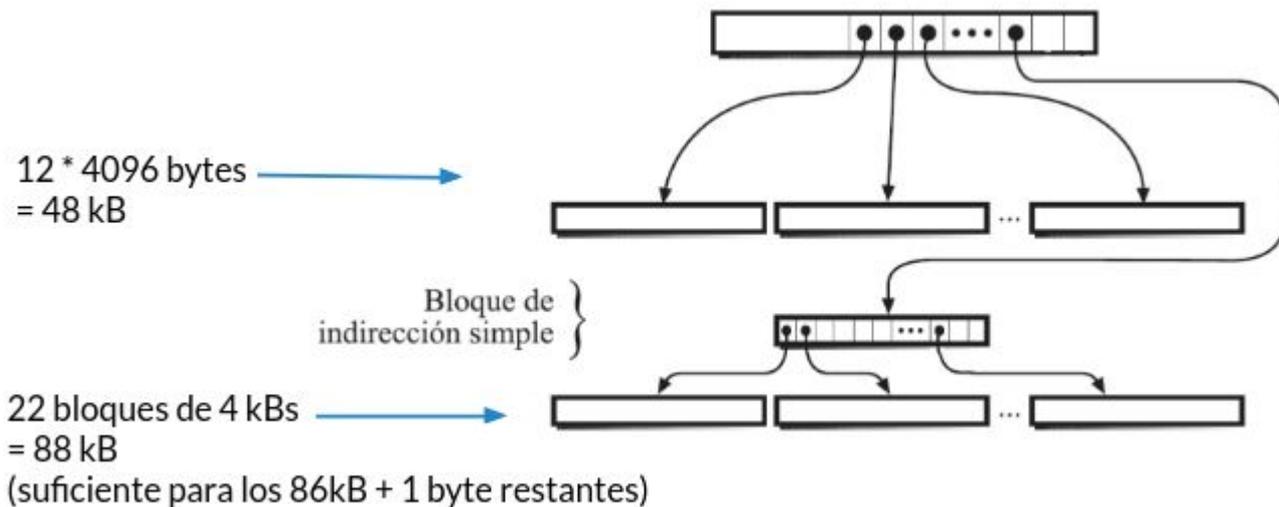
- a) Explique cuántos bloques de datos y bloques de indirección se utilizarán, suponiendo que el tamaño de los bloques de la partición es **4096** bytes.
- Tamaño del archivo $(12 + 256) * 512 + 1$ Bytes = **134 kB + 1 Byte**
 - Cantidad de bloques: $\lceil (134 \text{ kB} + 1 \text{ Byte}) / 4 \text{ kB} \rceil = \mathbf{34}$
 - Punteros por bloque: $4096 / 4 = \mathbf{1024}$

Problema 3: Inodes y bloques de datos

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

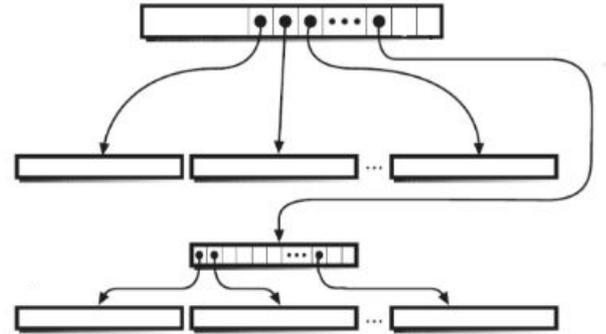
- a) Explique cuántos bloques de datos y bloques de indirección se utilizarán, suponiendo que el tamaño de los bloques de la partición es **4096** bytes.
- Tamaño del archivo $(12 + 256) * 512 + 1$ Bytes = **134 kB + 1 Byte**
 - Cantidad de bloques: $\lceil (134 \text{ kB} + 1 \text{ Byte}) / 4 \text{ kB} \rceil = \mathbf{34}$
 - Punteros por bloque: $4096 / 4 = \mathbf{1024}$
 - Utilizando los 12 punteros de datos que hay en el inodo podremos referenciar **$12 * 4096$ Bytes = 48 kB. Nos faltan 88kB.**
 - Con el puntero de indirección simple podremos referenciar hasta **$1024 * 4096$ Bytes = 4096 kB**, lo que es suficiente para contener los 86 kB + 1 Byte restantes (deberemos ocupar y referenciar 22 bloques de datos)

Problema 3: Inodes y bloques de datos



Problema 3: Inodes y bloques de datos

b) 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.

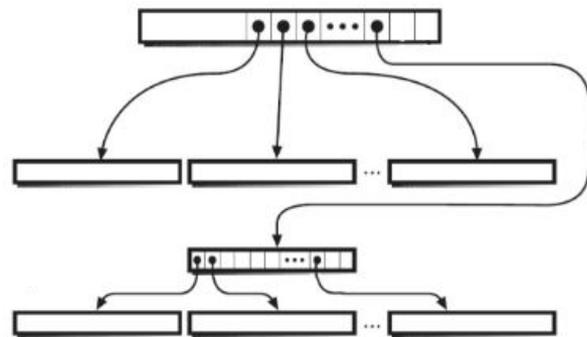


Problema 3: Inodes y bloques de datos

b) 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
- El bloque de datos.

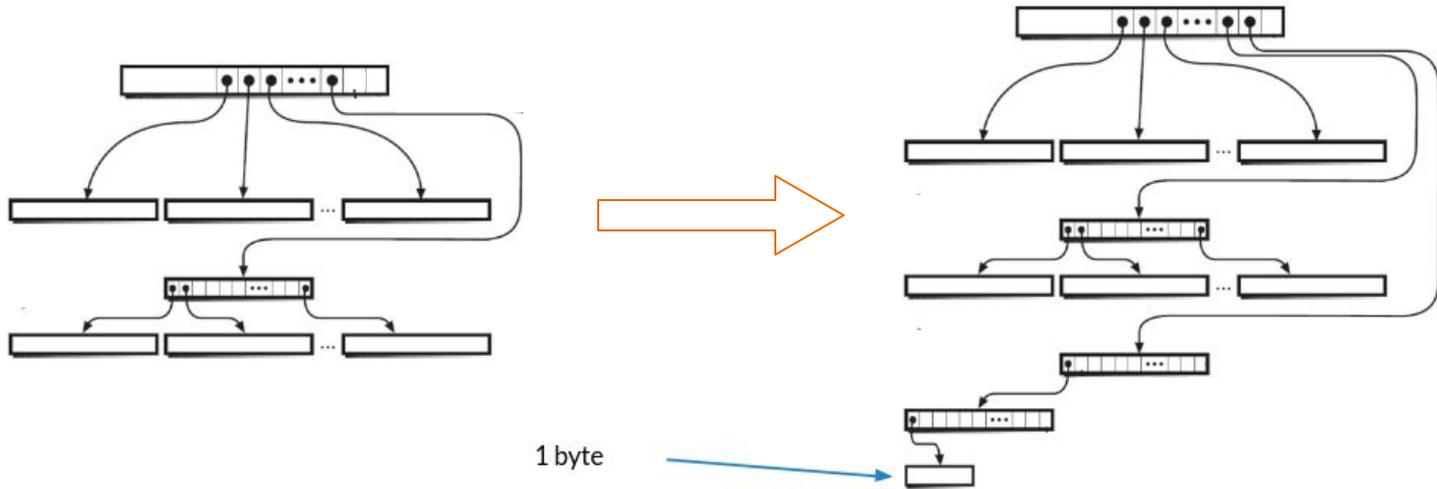


Problema 4: Inodes y bloques de datos

a) 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?

Problema 4: Inodes y bloques de datos

a) 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?



Problema 4: Inodes y bloques de datos

a) 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. Calculamos la cantidad de punteros en cada bloque:

$$\# \text{ punteros} = 2\text{kB} / 4\text{B} = 512$$

Problema 4: Inodes y bloques de datos

a) 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. Calculamos la cantidad de punteros en cada bloque:

$$\# \text{ punteros} = 2\text{kB} / 4\text{B} = 512$$

Considerando que se encuentran llenos todos los bloques que se puede direccionar usando el bloque de indirección simple:

- Inodo: **12 * 2kB**
- Bloque de indirección simple: **512 * 2kB**
- **+1 Byte** (añadido)

Problema 4: Inodes y bloques de datos

a) 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. Calculamos la cantidad de punteros en cada bloque:

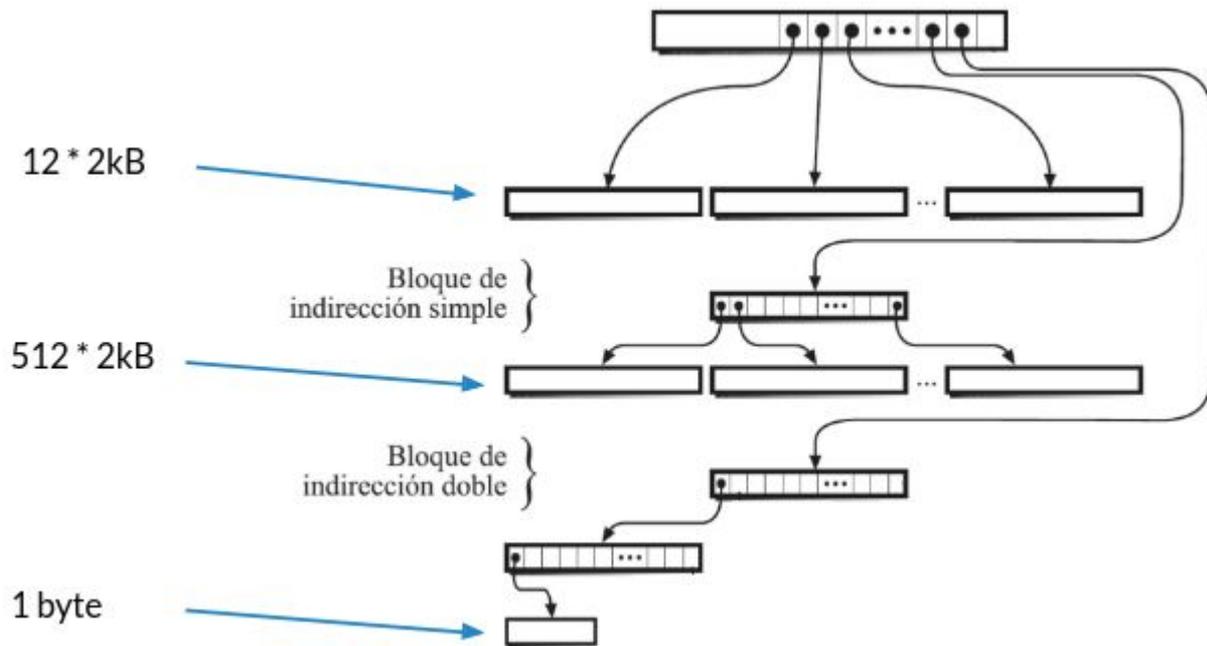
$$\# \text{ punteros} = 2\text{kB} / 4\text{B} = 512$$

Considerando que se encuentran llenos todos los bloques que se puede direccionar usando el bloque de indirección simple:

- Inodo: **12 * 2kB**
- Bloque de indirección simple: **512 * 2kB**
- **+1 Byte** (añadido)

Por lo tanto el archivo es de **12 * 2kB + 512 * 2kB + 1B = 1048kB + 1B.**

Problema 4: Inodes y bloques de datos



Archivos y directorios

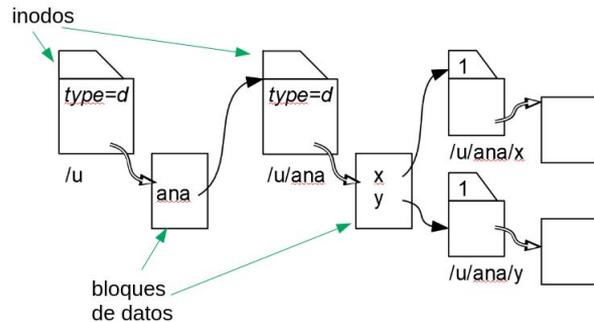
Archivos y directorios

- Un directorio ocupa un inodo y bloques de datos como los archivos normales
- Por cada archivo perteneciente a ese directorio se almacena una fila de largo variable
 - El primer byte indica el largo del nombre del archivo en bytes
 - Luego viene el nombre
 - Finalmente 2 bytes que indican el número del inodo que representa ese archivo
 - El nombre se almacena en el directorio, no en el inodo.
 - Un directorio siempre contiene un link duro a sí mismo (.) y un link duro del directorio padre (..)
- Por ejemplo si un directorio contiene los archivos dat.txt y hello:

1	.	inodo	
2	..	inodo'	
7	d a t . t x t		inodo"
5	h e l l o		inodo'''

Problema 5: Archivos y directorios

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

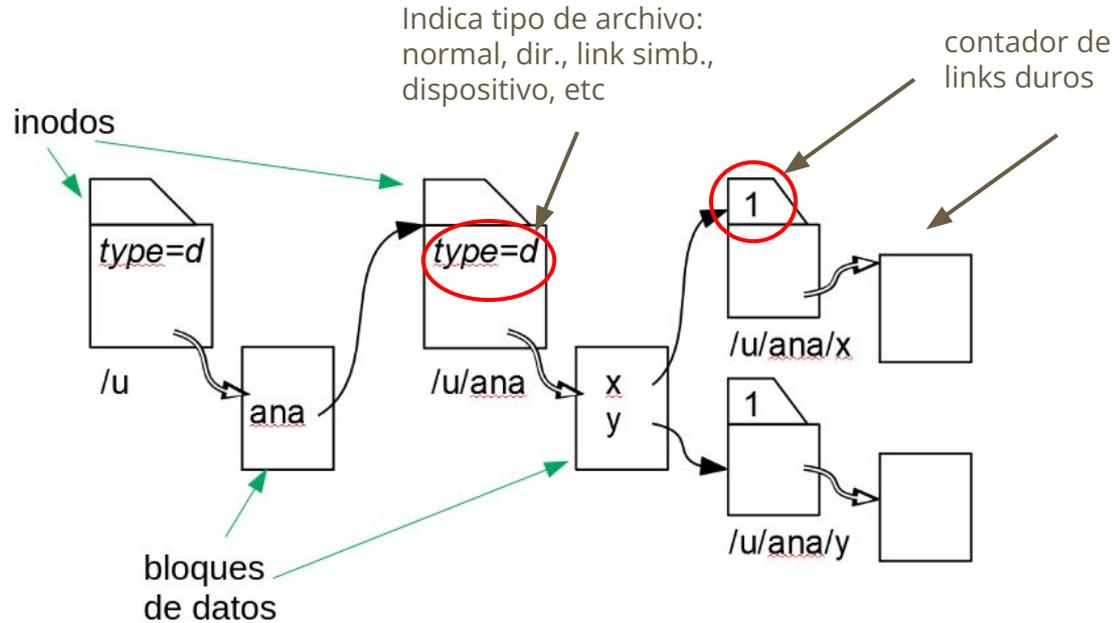


La usuaria ana ejecuta los siguientes comandos:

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

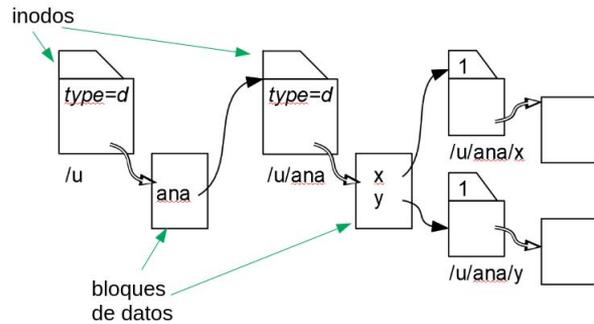
Rehaga la figura de acuerdo a los cambios realizados.

Archivos y directorios



Problema 5: Archivos y directorios

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



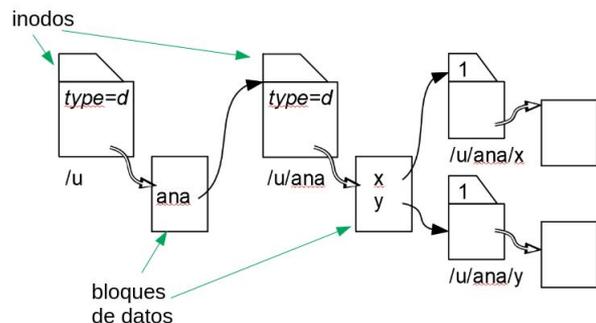
La usuaria ana ejecuta los siguientes comandos:

```
$ 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.

Problema 5: Archivos y directorios

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

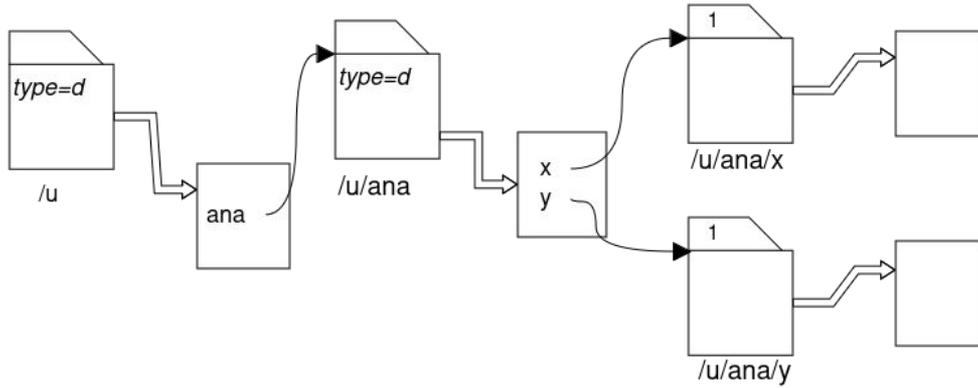


La usuaria ana ejecuta los siguientes comandos:

```
$ mkdir /u/ana/tmp      crea directorio
$ cd /u/ana             se mueve a dir /u/ana
$ ln x tmp/z            link a x en tmp/z
$ cp y tmp/w            copia y en tmp/w
$ cd tmp                se mueve a dir tmp
$ ln -s ../x v          link simbólico a ../x en v
```

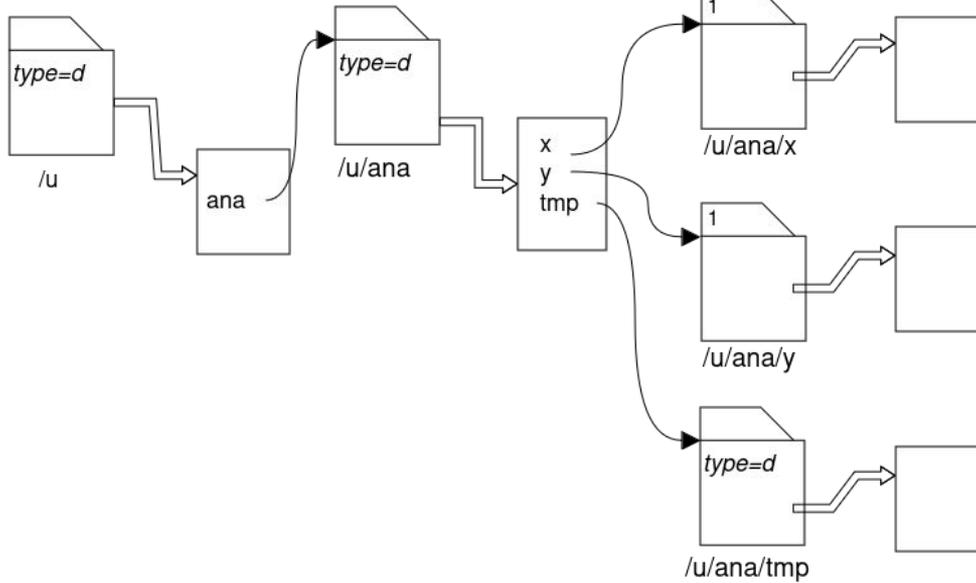
Rehaga la figura de acuerdo a los cambios realizados.

Problema 5: Archivos y directorios



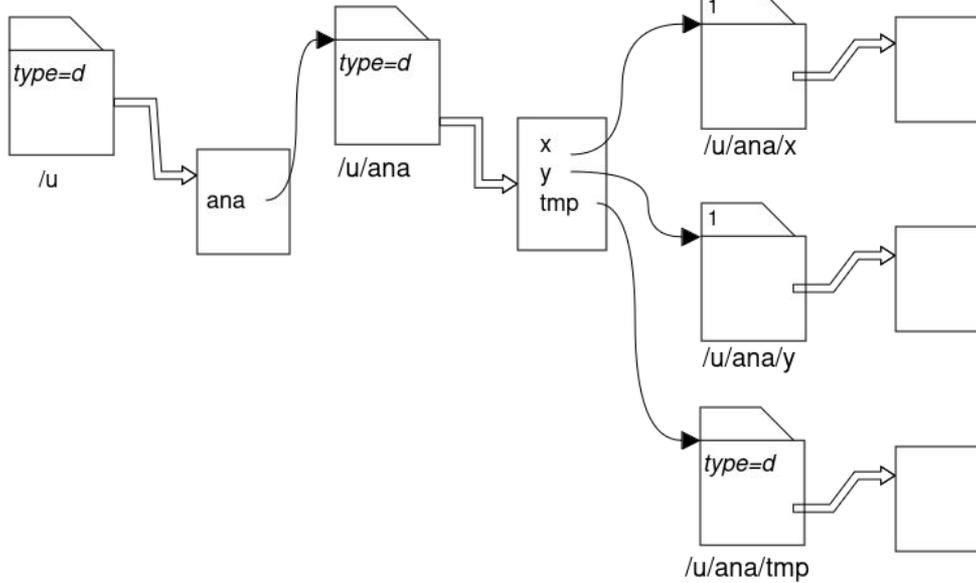
```
$ mkdir /u/ana/tmp  
$ cd /u/ana  
$ ln x tmp/z  
$ cp y tmp/w  
$ cd tmp  
$ ln -s ../x v
```

Problema 5: Archivos y directorios



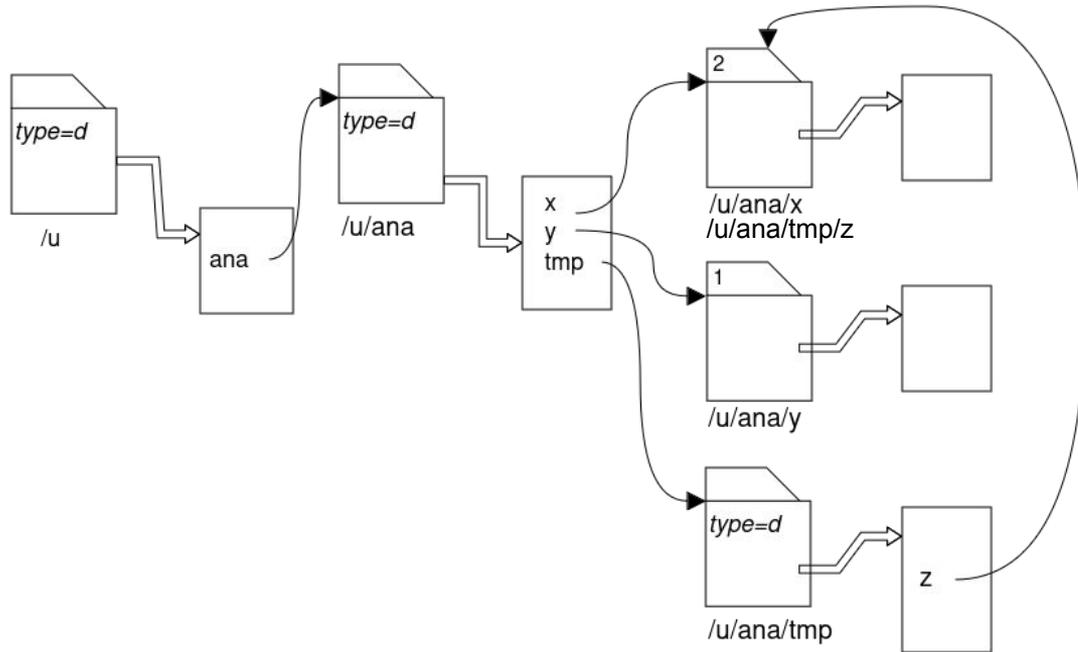
```
$ mkdir /u/ana/tmp  
$ cd /u/ana  
$ ln x tmp/z  
$ cp y tmp/w  
$ cd tmp  
$ ln -s ../x v
```

Problema 5: Archivos y directorios



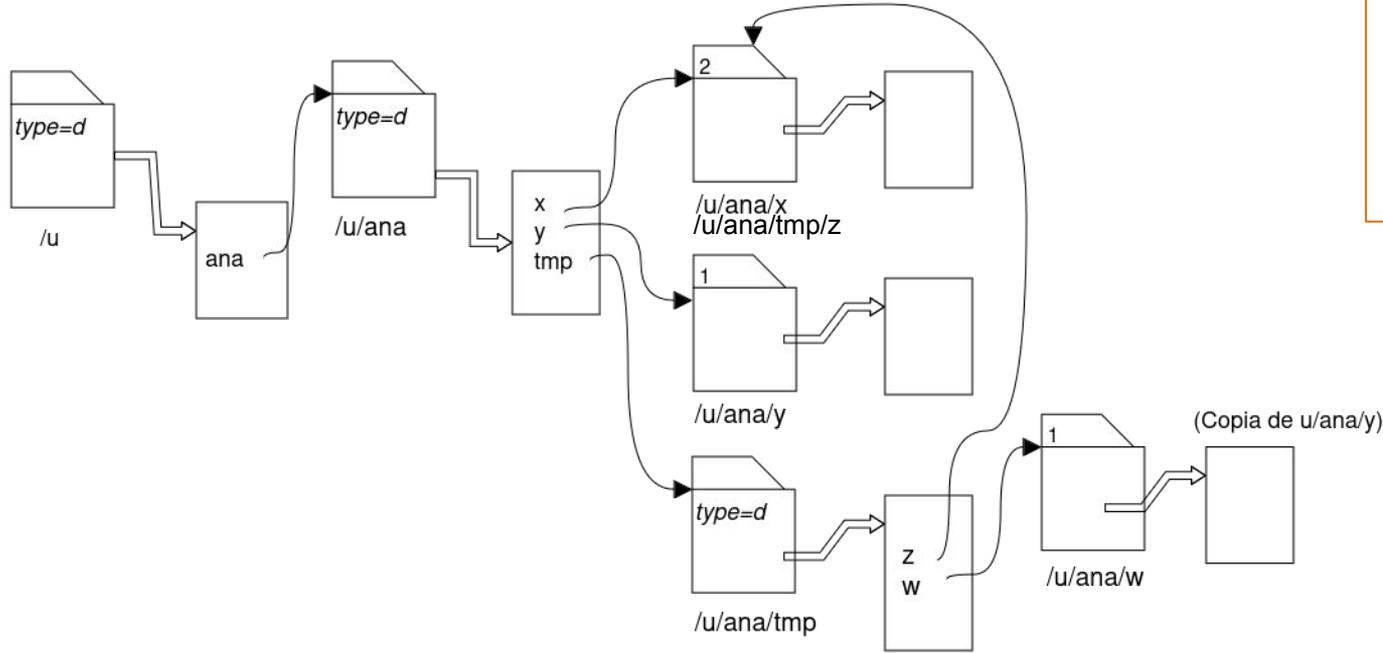
```
$ mkdir /u/ana/tmp  
$ cd /u/ana  
$ ln x tmp/z  
$ cp y tmp/w  
$ cd tmp  
$ ln -s ../x v
```

Problema 5: Archivos y directorios



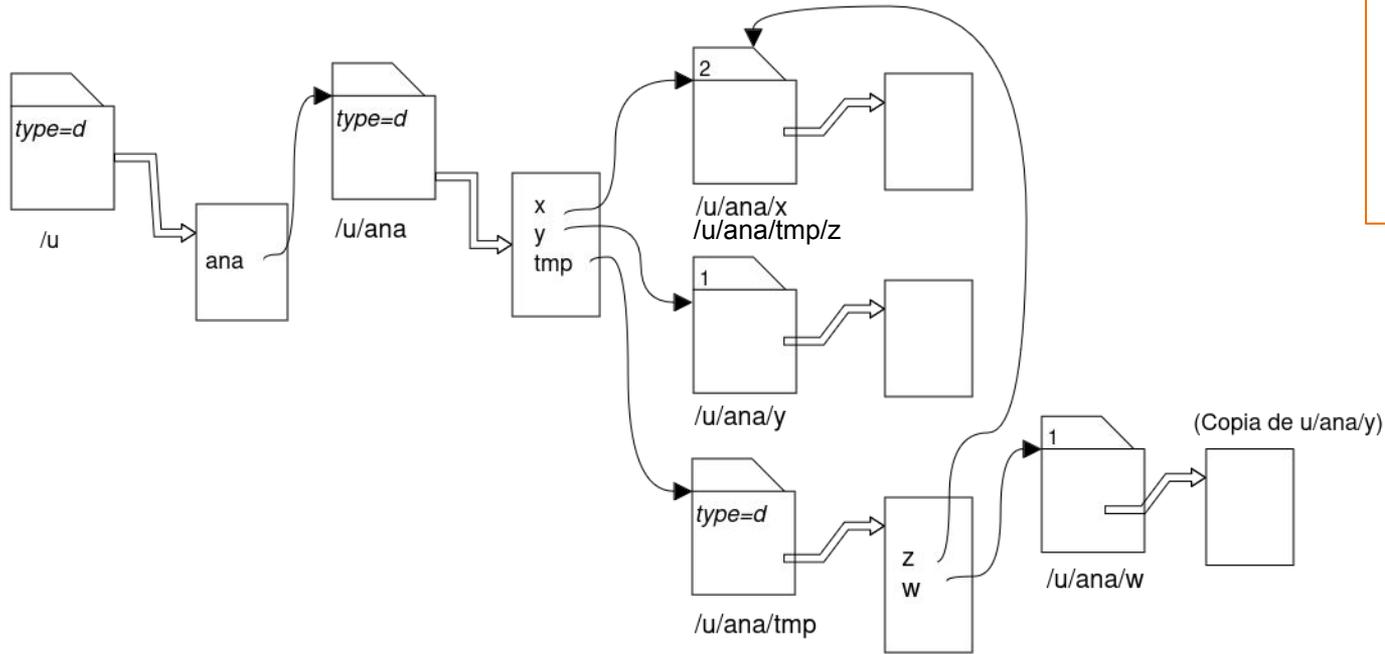
```
$ mkdir /u/ana/tmp
$ cd /u/ana
$ ln x tmp/z
$ cp y tmp/w
$ cd tmp
$ ln -s ../x v
```

Problema 5: Archivos y directorios



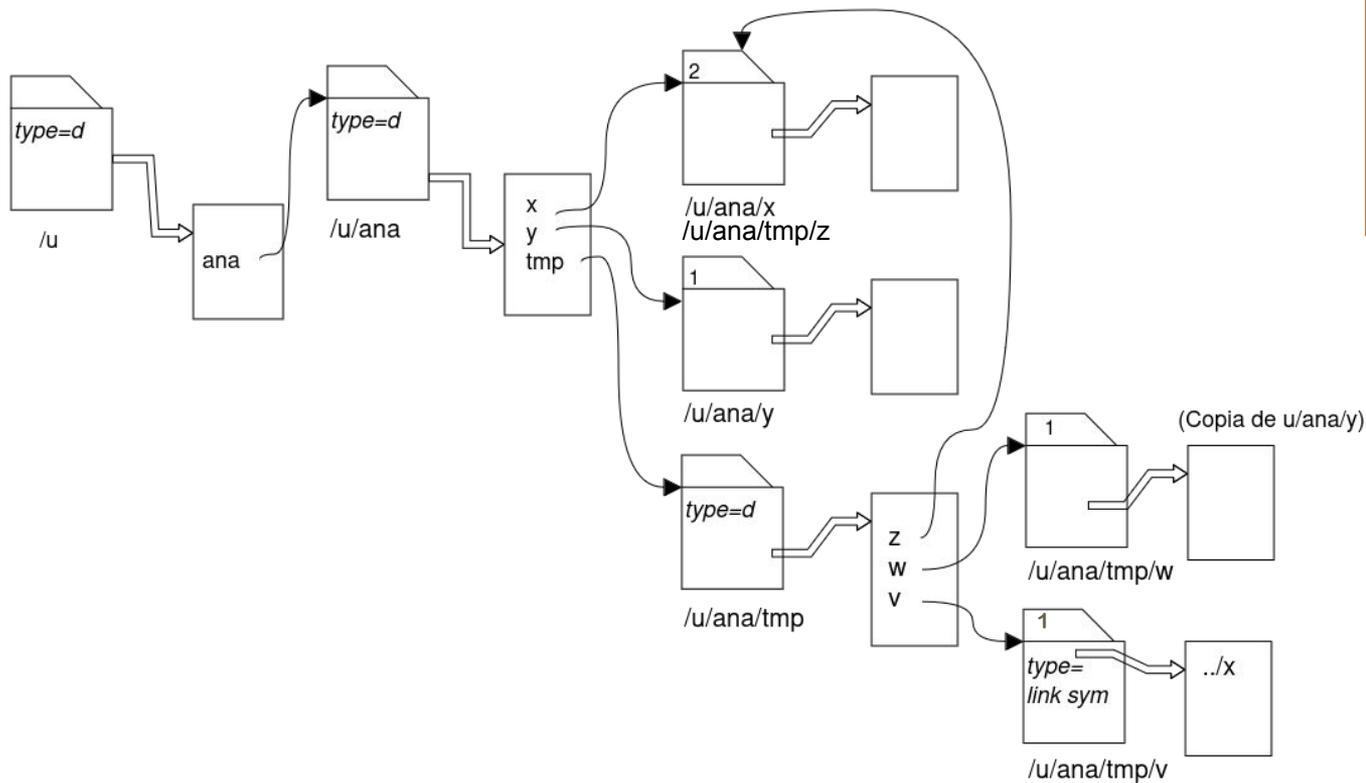
```
$ mkdir /u/ana/tmp
$ cd /u/ana
$ ln x tmp/z
$ cp y tmp/w
$ cd tmp
$ ln -s ../x v
```

Problema 5: Archivos y directorios



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

Problema 5: Archivos y directorios



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

Disco moderno vs SSD

Problema 6: 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:

Problema 6: 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:

Al ser dos archivos, se accede de manera alternada a estos y se encuentran en partes opuestas del disco: ¡Es importante considerar el tiempo de acceso a cada bloque!

Problema 6: Disco moderno vs SSD

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

Problema 6: 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.)

Problema 6: Disco moderno vs SSD

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

Problema 6: Disco moderno vs SSD

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

R. **read-ahead:** Lee más bloques que los solicitados porque es probable que se soliciten pronto

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

Problema 6: Disco moderno vs SSD

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

Problema 6: Disco moderno vs SSD

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

R. Considerando 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.

FIN