



OPERACIONES DE UN PROCESO SOBRE UN FICHERO

Pedro de Miguel Anasagasti

En esta animación se muestra el resultado obtenido al realizar un proceso una serie de operaciones sobre un fichero.

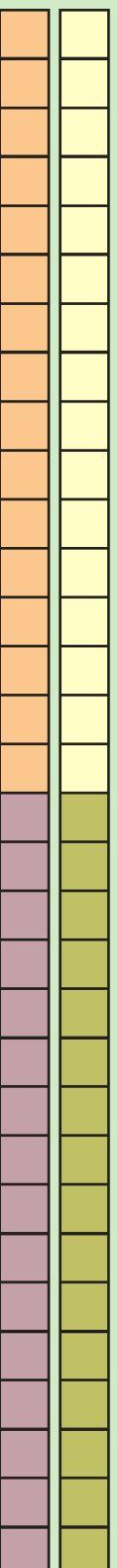
Para que los dibujos sean visibles vamos a considerar que el sector es de 8 bytes



el bloque de 2 sectores = 16 bytes



y la agrupación de 4 bloques = 64 bytes



(Sabemos que el sector real tiene un tamaño de entre 128 a 1024 bytes)

Se supone que existe una cache de bloques con política “**write through**”



Preparación

- Se selecciona la primera entrada libre en la tabla fd del proceso (7).
- Se selecciona la primera entrada libre de la tabla intermedia (4).
- Se selecciona un nodo-i libre (678).

Disco

Memoria



- $d \leftarrow 7$.
- Se rellenan adecuadamente las entradas de la tabla fd , de la tabla intermedia y el nodo- i .
- El fichero de momento no tiene espacio, no tiene asignada ninguna agrupación y su tamaño es 0 bytes.

SO

𐤀	𐤁	𐤂	𐤃	𐤄	𐤅	𐤆	𐤇	𐤈	𐤉
𐤊	𐤋	𐤌	𐤍	𐤎	𐤏	𐤐	𐤑	𐤒	𐤓
𐤔	𐤕	𐤖	𐤗	𐤘	𐤙	𐤚	𐤛	𐤜	𐤝
𐤞	𐤟	𐤠	𐤡	𐤢	𐤣	𐤤	𐤥	𐤦	𐤧
𐤨	𐤩	𐤪	𐤫	𐤬	𐤭	𐤮	𐤯	𐤰	𐤱
𐤲	𐤳	𐤴	𐤵	𐤶	𐤷	𐤸	𐤹	𐤺	𐤻
𐤼	𐤽	𐤾	𐤿	𐥀	𐥁	𐥂	𐥃	𐥄	𐥅
𐥆	𐥇	𐥈	𐥉	𐥊	𐥋	𐥌	𐥍	𐥎	𐥏
𐥐	𐥑	𐥒	𐥓	𐥔	𐥕	𐥖	𐥗	𐥘	𐥙
𐥚	𐥛	𐥜	𐥝	𐥞	𐥟	𐥠	𐥡	𐥢	𐥣
𐥤	𐥥	𐥦	𐥧	𐥨	𐥩	𐥪	𐥫	𐥬	𐥭
𐥮	𐥯	𐥰	𐥱	𐥲	𐥳	𐥴	𐥵	𐥶	𐥷
𐥹	𐥺	𐥻	𐥼	𐥽	𐥾	𐥿	𐦀	𐦁	𐦂
𐦃	𐦄	𐦅	𐦆	𐦇	𐦈	𐦉	𐦊	𐦋	𐦌
𐦍	𐦎	𐦏	𐦐	𐦑	𐦒	𐦓	𐦔	𐦕	𐦖
𐦗	𐦘	𐦙	𐦚	𐦛	𐦜	𐦝	𐦞	𐦟	𐦠
𐦡	𐦢	𐦣	𐦤	𐦥	𐦦	𐦧	𐦨	𐦩	𐦪
𐦫	𐦬	𐦭	𐦮	𐦯	𐦰	𐦱	𐦲	𐦳	𐦴
𐦵	𐦶	𐦷	𐦸	𐦹	𐦺	𐦻	𐦼	𐦽	𐦾
𐦿	𐧀	𐧁	𐧂	𐧃	𐧄	𐧅	𐧆	𐧇	𐧈
𐧉	𐧊	𐧋	𐧌	𐧍	𐧎	𐧏	𐧐	𐧑	𐧒
𐧓	𐧔	𐧕	𐧖	𐧗	𐧘	𐧙	𐧚	𐧛	𐧜
𐧝	𐧞	𐧟	𐧠	𐧡	𐧢	𐧣	𐧤	𐧥	𐧦
𐧧	𐧨	𐧩	𐧪	𐧫	𐧬	𐧭	𐧮	𐧯	𐧰
𐧱	𐧲	𐧳	𐧴	𐧵	𐧶	𐧷	𐧸	𐧹	𐧺
𐧻	𐧼	𐧽	𐧾	𐧿	𐨀	𐨁	𐨂	𐨃	𐨄
𐨅	𐨆	𐨇	𐨈	𐨉	𐨊	𐨋	𐨌	𐨍	𐨎
𐨏	𐨐	𐨑	𐨒	𐨓	𐨔	𐨕	𐨖	𐨗	𐨘
𐨙	𐨚	𐨛	𐨜	𐨝	𐨞	𐨟	𐨠	𐨡	𐨢
𐨣	𐨤	𐨥	𐨦	𐨧	𐨨	𐨩	𐨪	𐨫	𐨬
𐨭	𐨮	𐨯	𐨰	𐨱	𐨲	𐨳	𐨴	𐨵	𐨶
𐨷	𐨸	𐨹	𐨺	𐨻	𐨼	𐨽	𐨾	𐨿	𐩀
𐩁	𐩂	𐩃	𐩄	𐩅	𐩆	𐩇	𐩈	𐩉	𐩊
𐩋	𐩌	𐩍	𐩎	𐩏	𐩐	𐩑	𐩒	𐩓	𐩔
𐩕	𐩖	𐩗	𐩘	𐩙	𐩚	𐩛	𐩜	𐩝	𐩞
𐩟	𐩠	𐩡	𐩢	𐩣	𐩤	𐩥	𐩦	𐩧	𐩨
𐩩	𐩪	𐩫	𐩬	𐩭	𐩮	𐩯	𐩰	𐩱	𐩲
𐩳	𐩴	𐩵	𐩶	𐩷	𐩸	𐩹	𐩺	𐩻	𐩼
𐩽	𐩾	𐩿	𐪀	𐪁	𐪂	𐪃	𐪄	𐪅	𐪆
𐪇	𐪈	𐪉	𐪊	𐪋	𐪌	𐪍	𐪎	𐪏	𐪐
𐪑	𐪒	𐪓	𐪔	𐪕	𐪖	𐪗	𐪘	𐪙	𐪚
𐪛	𐪜	𐪝	𐪞	𐪟	𐪠	𐪡	𐪢	𐪣	𐪤
𐪥	𐪦	𐪧	𐪨	𐪩	𐪪	𐪫	𐪬	𐪭	𐪮

2	4
---	---

Tabla intermedia			
Nodo-i Puntero Referen. rw			
463	37485	2	01

Agrupac.:
Tipo: normal
Tamaño: 0

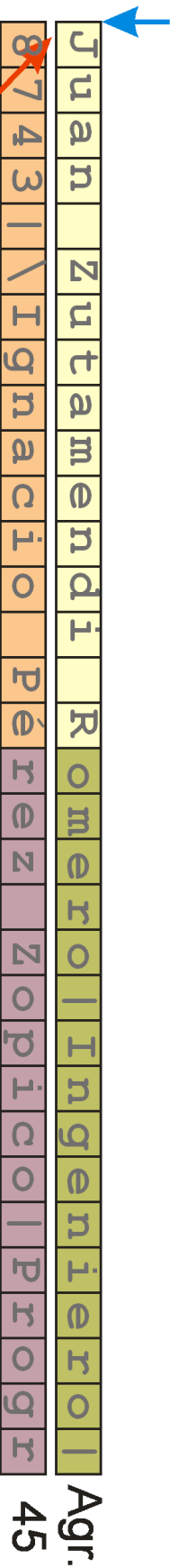
6

escrito = write(d, "abcdefghijklmnpqrstuvwxyz", 26);

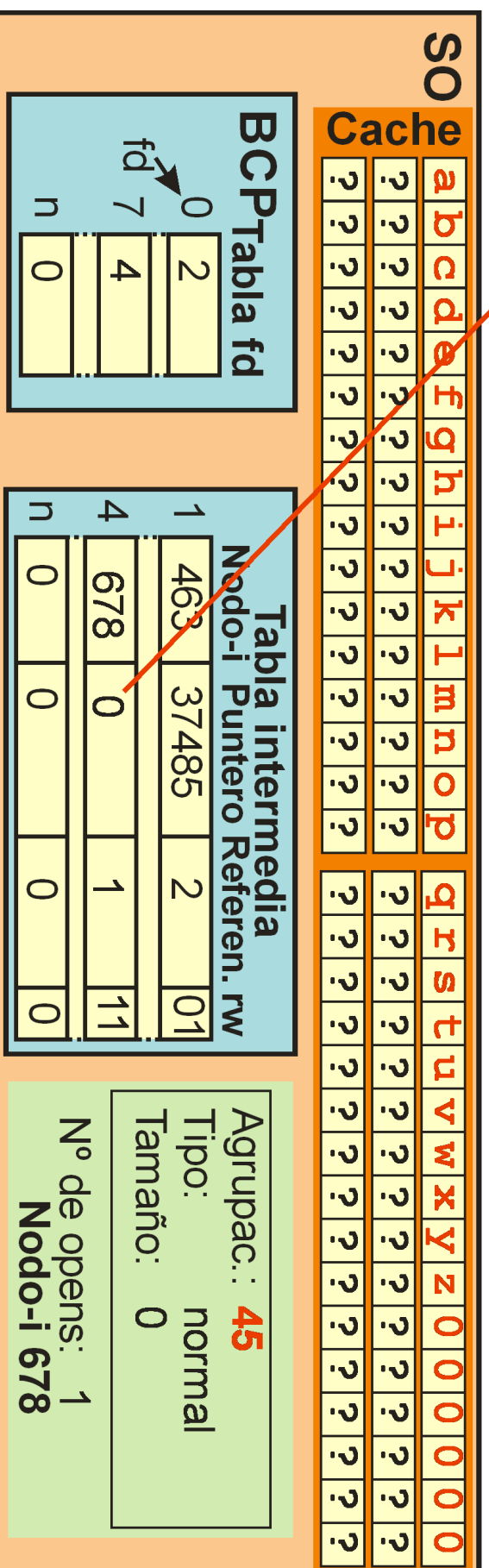
Preparación

- El 'write' se hace a través del descriptor d = 7.
- No hay espacio $\Rightarrow$ se asigna una agrupación (p.e. la 45). Tiene basura.
- Se seleccionan dos bloques de la cache y se rellenan.
- Marcaremos con  el final del fichero.

Disco



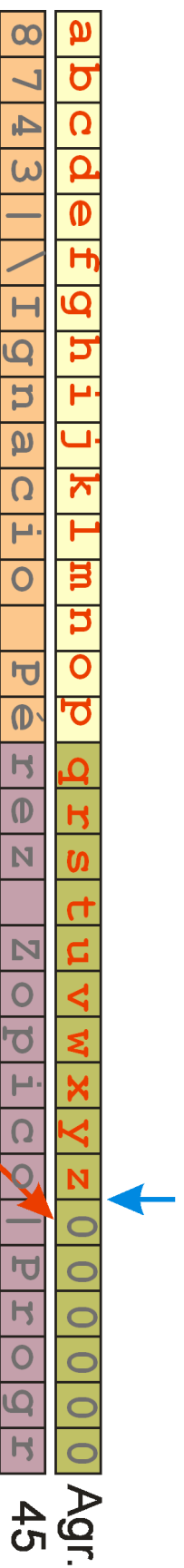
Memoria



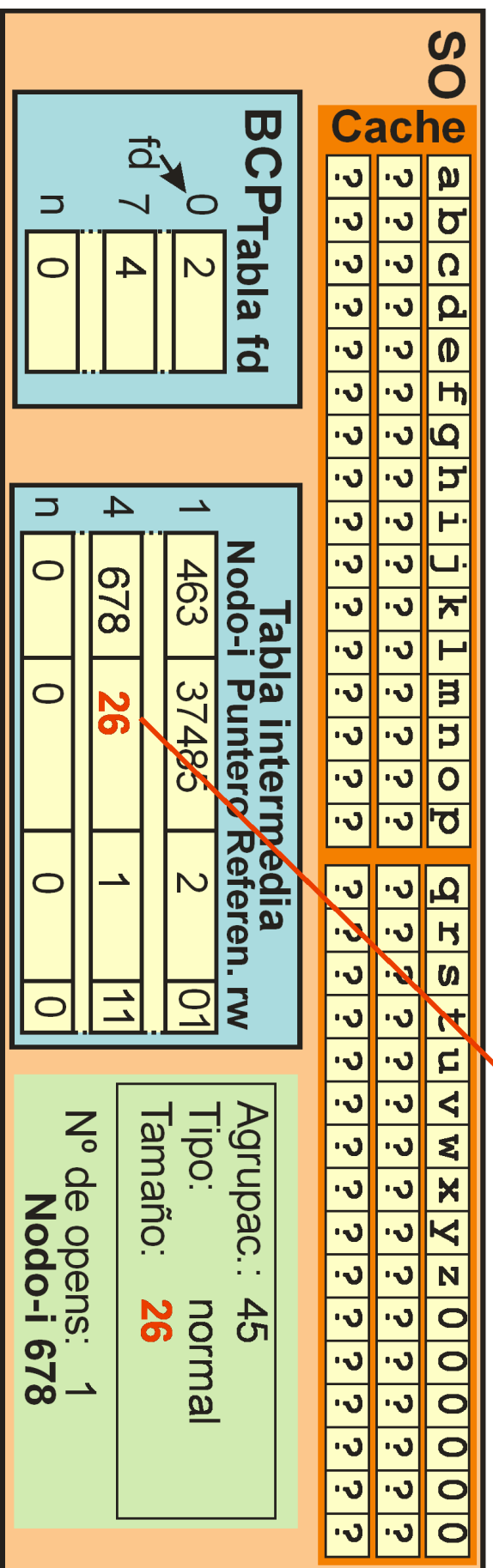
Ejecución

- ## Ejecución

Disco



Memoria



leido=read(d, buf, 26);

Preparación

Disco

a b c d e f g h i j k l m n o p q r s t u v w x y z 0 0 0 0 0 0 Agr. 45

8 7 4 3 1 \ I g n a c i o P e r e z Z o p i c o l P r o g r



Disco

SO

Cache

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

BCP Tabla fd

0	2
7	4
n	0

Tabla intermedia
Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	26	1	11
n	0	0	0	0

Agrupac.: 45

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i 678

- El 'read' se hace a través del descriptor $d = 7$.
- No se lee nada, puesto que estamos al final del fichero.
- $leido \leftarrow 0$.
- buf queda sin modificar.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0	
8	7	4	3	1	\	I	g	n	a	c	i	o		P	e	r	e	z		Z	o	p	i	c	c	o		P	r	o	g	r

Agr.

45

SO

Cache

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

BCP Tabla fd

0	2		
fd 7	4		
n	0		

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2 01
4	678	26	1 11
n	0	0	0 0

Agrupac. : 45

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i 678

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0	Agri.
8	7	4	3		/	I	g	n	a	c	i	o		P	e	r	e	z		z	o	p	i	c	c		P	r	o	g	r	45

So

[illegible]

BCP-Tabla fd

0	7	n
2	4	0

Tabla intermedia
Nodo-i **Puntero** **Referen. rw**

Nº de opens: 1

lseek(d, 1, SEEK_SET);

Ejecución

- Puntero $\leftarrow 1$.

a b c d e f g h i j k l m n o p q r s t u v w x y z 0 0 0 0 0 0 Agr. 45

8 7 4 3 1 \ I g n a c i o P e r r e z Z o p i c o l P r o g r



Disco

SO

Cache	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

BCPTabla fd

0	2
fd 7	4
n	0

Tabla intermedia
Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	1	11	
n	0	0	0	0

Agrupac.: 45

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i 678

Memoria

escrito = write(d,"BCDE", 4);

Preparación

- Hay espacio suficiente para realizar la escritura.

Disco

a b c d e f g h i j k l m n o p q r s t u v w x y z 0 0 0 0 0 0 Agr. 45

8 7 4 3 1 \ I g n a c i o P é r e z Z o p i c o l P r o g r



Disco

SO

Cache

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

BCP Tabla fd

0	2						
fd	7	4					
n	0						

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2
4	678	1	1
n	0	0	0

Agrupac.: 45

Tipo: normal

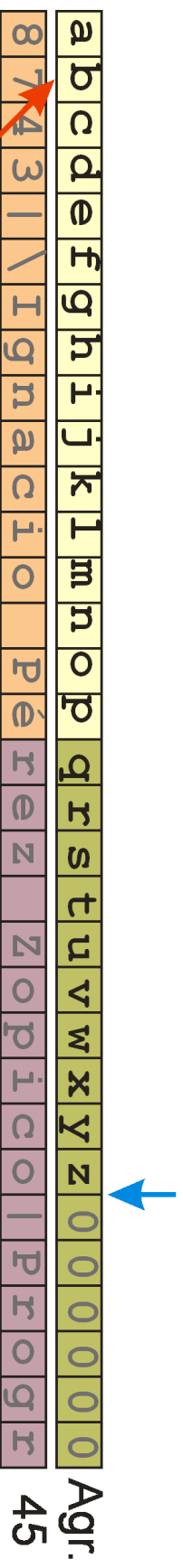
Tamaño: 26

Nº de opens: 1

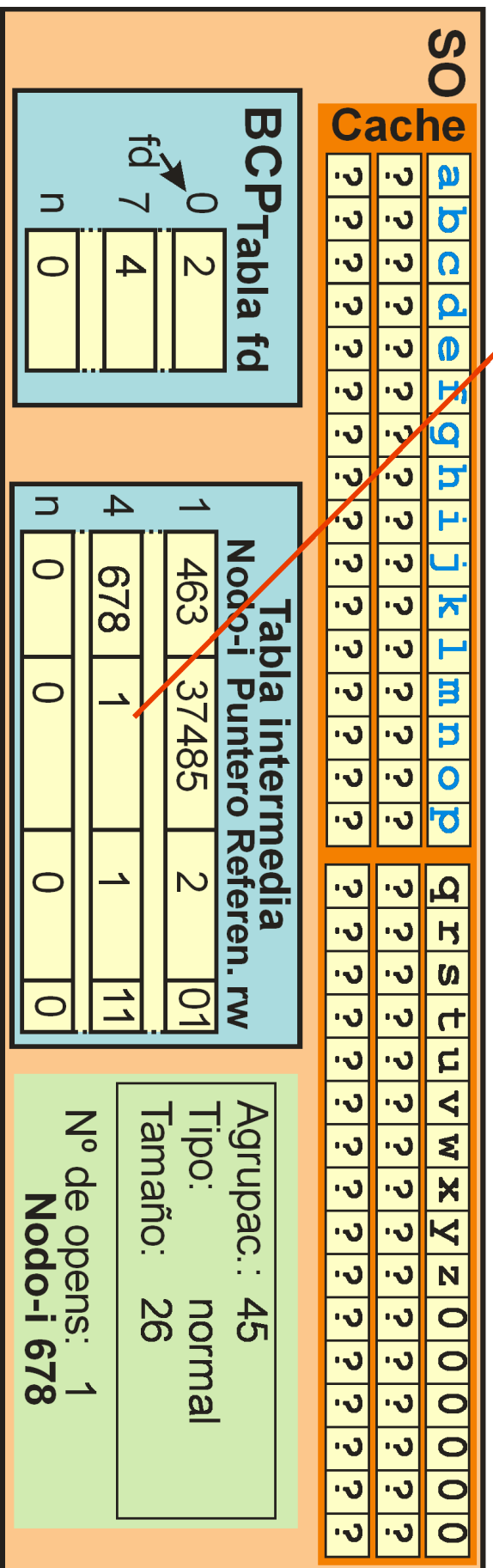
Nodo-i 678

- Como el bloque está en la cache de bloques, no hay que leerlo del disco, en caso de no estar habría que leerlo, quedando en la cache.

Disco



Memoria



- El sistema de ficheros modifica el bloque de la cache.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
8	7	4	3	1	\	I	g	n	a	c	i	o		P	e	r	e			Z	o	p	i	c	o		P	r	o	g	r

45 Agr.

SO

a	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

Cache

a	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	



Ejecución: se escribe el bloque completo

- La escritura sobrescribe, NO inserta. Se graba en el disco el bloque.
- escrito $\leftarrow$ 4.
- Puntero $\leftarrow$ 5.

a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	
8	7	4	3		/	I	g	n	a	c	i	o		P	e	r	e	n		Z	o	p	i	c	o		P	r	o	g	r
																												Agr.			
																												45			

Cache

a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

SO

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BCP

2
4
0

fd 0

fd 7

n

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2 01
4	678	5	1 11
n	0	0	0 0

Agrupac.: 45

Tipo: normal

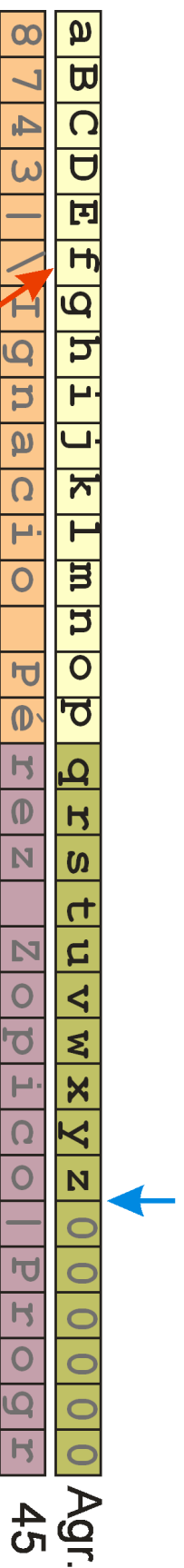
Tamaño: 26

Nº de opens: 1

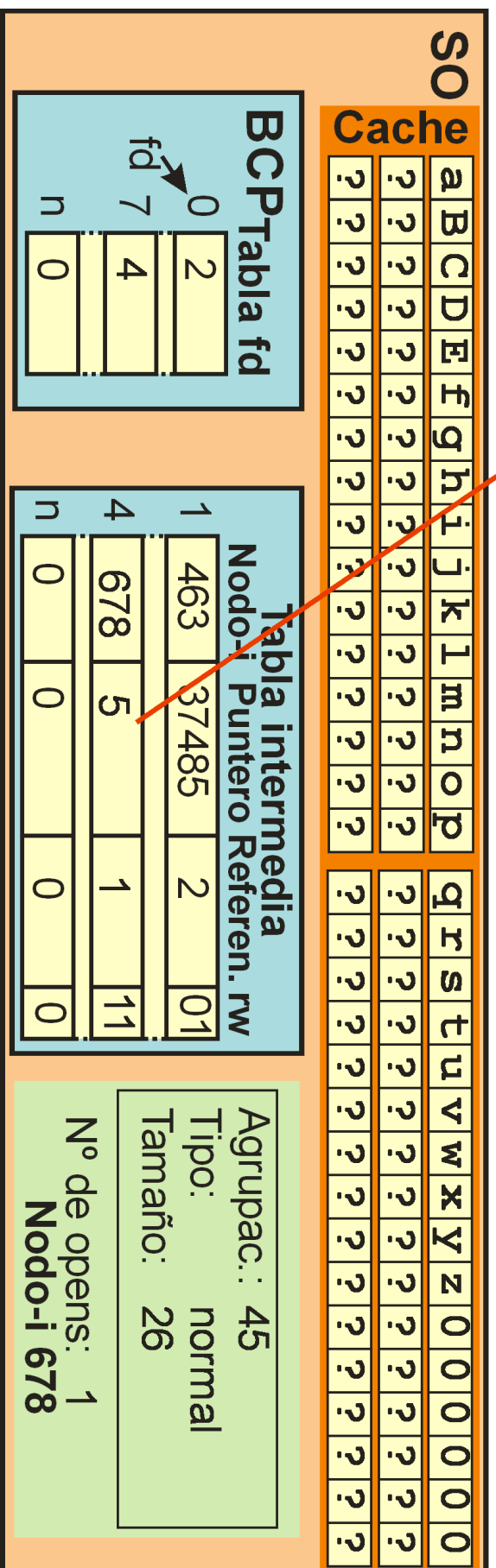
Nodo-i 678

- Com el bloque està en la cache de blocques, el sistema de fitxers no necessita **leer** el bloque del disc. En cas contrari se **leería** el bloque y se **dejaría** en la cache de blocques.

Disco



Memoria



leido=read(d, buf, 5);

Ejecución: se usa la parte solicitada

- leido ← 5.
- Puntero ← 10.
- buf ← fghij.

a B C D E f g h i j k l m n o p q r s t u v w x y z 0 0 0 0 0 0 Agr. 45

8 7 4 3 1 \ I g n a c i o P e r r e z Z o p i c o l P r o g r



Disco

SO																									
Cache																									
a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Memoria

BCP Tabla fd			
0	2		
fd 7	4		
n	0		

Tabla intermedia				
Nodo-i	Puntero	Referen.	rw	
1	463	37485	2	01
4	678	10	1	11
n	0	0	0	0

Agrupac.: 45
Tipo: normal
Tamaño: 26
Nº de opens: 1
Nodo-i 678

- No hay suficientes datos. El SO suministra los que hay.
- No hace falta leer los bloque del disco, puesto que están en la cache de bloques.

a	B	C	D	E	F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0	
8	7	4	3	1	\	/	I	g	n	a	c	i	o			P	e	r	e	z		Z	O	P	I	C	O	I	P	r	O	g	r
																												Agr.	45				

SO

Cache

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

BCP

Tabla fd

0	2	
7	4	
n	0	

Tabla intermedia

Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	10	1	11
n	0	0	0	0

Agrupac.: 45

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i 678

- leido $\leftarrow$ 16.
- Puntero $\leftarrow$ 26.
- buf $\leftarrow$ klmnopqrstuvwxyz.

a	B	C	D	E	F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
8	7	4	3	1	\		g	n	a	c	i	o		P	e	r	e	z		Z	o	p	i	c	c	o		P	r	o	g	r
																												Agr.	45			

SO

a	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

Cache

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

BCP-Tabla fd

0	2
fd ↗	4
n	0

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2 01
4	678	26	1 11
n	0	0	0 0

Agrupac.: 45
Tipo: normal
Tamaño: 26

Nº de opens: 1
Nodo-i 678

- El bloque que hay que modificar parcialmente ya está en la cache.
- No hay espacio suficiente, por lo que hay que asignar una nueva agrupación al fichero (p.e. la 73).
- La nueva agrupación conserva lo grabado anteriormente.

a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
Agr.																															
45																															
8	7	4	3		\	I	g	n	a	c	i	o		P	e	r	e	z		Z	o	p	i	c	o		P	r	o	g	r
Agr.																															
7	3	6	5	2	8	4	6	4	6	3	7	2	9	8	9	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4
Agr.																															
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9
Agr.																															

Cache

a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

SO

q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2
4	678	26	1

BCPTabla fd

0	2
7	4
n	0

Agrupac.: 45, **73**

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i **678**

Preparación

- Se sobrescribe el primer bloque en la cache de bloques.
- Se seleccionan otros tres bloques de la cache de bloques y se rellenan con los datos a escribir.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	0	0	0	0	0
Agr.																															
45																															
8	7	4	3		\	I	g	n	a	c	i	o		P	e	r	e	z		Z	o	p	i	c	c		P	r	o	g	r
Agr.																															
73																															
7	3	6	5	2	8	4	6	4	6	3	7	2	9	8	9	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4
Agr.																															
73																															
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9
Agr.																															
73																															

Memoria

SO

Cache

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

BCP

Tabla fd

0

2

fd

7

4

n

0

Tabla intermedia

Nodo-i Puntero Referen. rw

1	463	37485	2		01
4	678	26	1		11
n	0	0	0		0

Agrupac.: 45, 73

Tipo: normal

Tamaño: 26

Nº de opens: 1

Nodo-i 678

Ejecución: se escriben en el disco los 4 bloques afectados.

- escrito $\leftarrow$ 40.
- Puntero $\leftarrow$ 66.
- Tamaño $\leftarrow$ 66.

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9
Agr. 45																															
Agr. 73																															

[illegible]

Cache																															
a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

BCP_{Table} fd

fd	n
0	2
7	4
n	0

Tabla intermedia				
Nodo-i Puntero Referen. rw				
1	463	37485	2	01
4	678	66	1	11
n	0	0	0	0

Agrupac.: 45, 73
Tipo: normal
Tamaño: 66

lseek(d, 50, SEEK_CUR);

Preparación

- Se hace un 'lseek' más allá del final del fichero.
- Si se escribe a partir de dicho punto se creará un agujero de 50 bytes.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4	Agr.
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9	73

SO

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Cache

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Memoria

BCP

Tabla fd	fd
0	2
7	4
n	0

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2
4	678	66	1
n	0	0	0

Agrupac.: 45, 73
Tipo: normal
Tamaño: 66
Nº de opens: 1
Nodo-i 678

lseek(d, 50, SEEK_CUR);

Ejecución

- Puntero $\leftarrow 116$.
- El tamaño NO aumenta
- El SO debe prohibir el acceso al hueco. Cuando éste se haga efectivo tendrá que borrar su contenido, p.e. rellenándolo con 'ceros'.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4	Agr.
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9	73

SO

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Cache

9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BCP Tabla fd

0	2				
fd 7	4				
n	0				

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2
4	678	116	1
n	0	0	0

Agrupac.: 45, 73

Tipo: normal

Tamaño: 66

Nº de opens: 1

Nodo-i 678

Memoria

texto = "Esto es la vida";
 escrito = write(d, texto, 6);



Preparación

- Se crea el agujero. El tamaño aumenta en el agujero + lo escrito
- Se selecciona un bloque de la cache y se rellena
- El SO debe borrar el contenido del agujero existente, p.e. rellenándolo con 'ceros'. Estos bloque no se guardan en la cache

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	7	2	6	2	*	7	6	2	5	5	1	5	1	4	4	Agr.
8	7	6	2	5	3	2	6	7	2	8	9	2	0	1	8	1	7	6	1	5	1	4	1	2	7	6	8	9	2	0	9	73

SO

Cache																															
a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	o	o	o	o	o	o	o

Cache

a	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

BCP Tabla fd

0	2
7	4
n	0

Tabla intermedia

Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	116	1	11
n	0	0	0	0

Agrupac.: 45, 73

Tipo: normal

Tamaño: **116**

Nº de opens: 1

Nodo-i 678

Memoria

texto = "Esto es la vida"
escrito = write(d, texto, 6);

Ejecución: Se han tenido que escribir tres bloques

- escrito $\leftarrow 6$.
- Puntero $\leftarrow 126$.
- Tamaño $\leftarrow 126$.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73

SO

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Cache

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Memoria

BCP Tabla fd

0	2
7	4
n	0

Tabla intermedia

Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	122	1	11
n	0	0	0	0

Agrupac.: 45, 73

Tipo: normal

Tamaño: 122

Nº de opens: 1

Nodo-i 678

lseek(d, 5000, SEEK_CUR);

Preparación

- Se hace un 'lseek' más allá del final del fichero.
- Si se escribe a partir de dicho punto se creará un agujero de 5000 bytes.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73

73

Disco

SO	Cache																															
	a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o		e	0	0	0	0	0	0	0

Memoria

BCP	Tabla	fd
0	2	
7	4	
n	0	

Tabla intermedia	Nodo-i	Puntero	Referen.	rw
1	463	37485	2	01
4	678	122	1	11
n	0	0	0	0

Agrupac.: 45, 73
Tipo: normal
Tamaño: 122
Nº de opens: 1
Nodo-i 678

lseek(d, 5000, SEEK_CUR);

Ejecución

- Puntero $\leftarrow 5106$.
- El SO debe prohibir el acceso al hueco. Cuando éste se haga efectivo tendrá que borrar su contenido, p.e. rellenándolo con 'ceros'.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73

SO

Cache																															
a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o		e	0	0	0	0	0	0	

Cache

9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BCP Tabla fd

0	2
fd 7	4
n	0

Tabla intermedia

Nodo-i	Puntero	Referen.	rw
1	463	37485	2
4	678	5122	1
n	0	0	0

Agrupac.: 45, 73

Tipo: normal

Tamaño: 122

Nº de opens: 1

Nodo-i 678

Memoria

close(d);

Preparación



Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	0	0	0	0	0	0	0	73

Disco

SO Cache	a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o		e	0	0	0	0	0	0	0

Memoria

BCP	Tabla	fd
0	2	
fd 7	4	
n	0	

Tabla intermedia				
	Nodo-i	Puntero	Referen.	rw
1	463	37485	2	01
4	678	5122	1	11
n	0	0	0	0

Agrupac.: 45, 73
Tipo: normal
Tamaño: 122
Nº de opens: 1
Nodo-i 678

close(d);

Ejecución

- Se libera la entrada 7 de la tabla de descriptors.
- Se decrementa el número de duplicados de la entrada 4 de la tabla intermedia. Como se queda en cero se libera dicha entrada.
- Se decrementa el número de opens del nodo-i. Al quedarse en 0 indica que no hay usuarios, por tanto el nodo-i en memoria se libera.



Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73

SO																															
Cache																															
a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	0	0	0	0	0	0	0

Memoria

BCPTabla fd				
0	2			
7	0			
n	0			

Tabla intermedia				
1	463	37485	2	01
4	678	5422	0	44
n	0	0	0	0

Agrupac.: 45, 73	
Tipo: normal	
Tamaño: 122	
Nº de opens: 0	Nodo-i 678

leído=read(d, buf, 26);

Preparación

- El descriptor d = 7 nos lleva a una entrada de la tabla fd no válida.

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	0	0	0	0	0	0	0	73

SO

Cache																															
a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o		e	0	0	0	0	0	0

Cache

a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	0	0	0	0	0	0	0	0

Memoria

BCPTabla fd

0	2		
fd 7	0		
n	0		

Tabla intermedia Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	5422	0	44
n	0	0	0	0

leido=read(d, buf, 26);

Ejecución

- El SO indica que la operación solicitada no es correcta.
- leido $\leftarrow -1$
- errno $\leftarrow$ EBADF (descriptor inválido).

Disco

a	B	C	D	E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6	Agr.
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	45
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Agr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o	e	0	0	0	0	0	0	0	73

SO

Cache																															
a	B	C	D	E	F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	s	t	o		e	0	0	0	0	0	0	0

Cache

a	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E	S	T	O		E	0	0	0	0	0	0	0

Memoria

BCPTabla fd

0	2		
fd 7	0		
n	0		

Tabla intermedia Nodo-i Puntero Referen. rw

1	463	37485	2	01
4	678	5422	0	44
n	0	0	0	0

FIN

