# EJEMPLO DE TRATAMIENTO DE ARCHIVOS EN SISTEMAS UNIX VERSION PRIMERA: LAS APERTURAS ESTAN ANTES DE FORK

### Antes de fork:

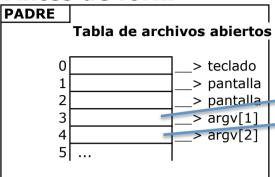


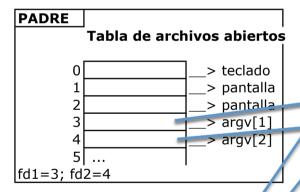
TABLA DE ARCHIVOS (global)				
contador	offset	modo		
			,	
1	1	RDONLY	> inodo en MP de argv[1]	
1	1	WRONLY	> inodo en MP de argv[2]	

OFFSET: PUNTERO DE LECTURA/ESCRITURA

	TABLA DE TNODOS (GIODAI)				
	contador	etc	_		
			_		
7	1		(argv[2])		
			_		
]			_		
]			_		
			-		
	1		(argv[1])		

TARLA DE INODOS (alabal)

### Tras fork:





contador	offset	modo	
			,
2	1	RDONLY	> inedo en MP de argv[1]
2	1	WRONLY	> inode en MP de argv[2]
1			

#### TABLA DE INODOS (global)

	contador	etc	- -
	1		_ _(argv[2])
]			-
]			-
	1		(argv[1])

```
PROGRAMA version primera:
char c[2]="\0\0";
fd1=open(argv[1], O_RDONLY))<0)
.... /// no presentamos todo el tratamiento de errores....
fd2=open(argv[2],O_CREAT|O_TRUNC|O_WRONLY)
.....
fork();
while (read(fd1,c,1)==1) {
        sleep( random()%4 /10);
        write ( fd2, c,1); }</pre>
```

# **VERSION SEGUNDA: LAS APERTURAS ESTAN TRAS FORK Tras las dos aperturas en los dos procesos:**

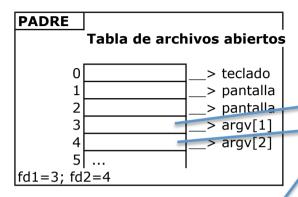
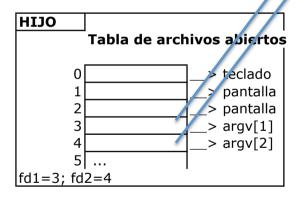


TABLA DE ARCHIVOS (global)			TABLA DE INODOS (global)			
contador	offset	modo		contador	etc	_
						_
					2	_(argv[2])
						_
1	1	RDONLY	> inodo en MP de argv[1]			_
. 1	1	WRONLY	> inode en MP de argv[2]			_
1	1	RDONLY	> imodo en MP de arqv[1]			_
1	1	WRONL	> inodo en MP de argv[2]	2	2	(argv[1])
						_

OFFSET: PUNTERO DE LECTURA/ESCRITURA



## PREGUNTAS QUE PODEMOS REALIZAR EN CADA VERSIÓN:

- 1) ¿Se generan dos copias o una sola copia de argv[1] en argv[2]?
- 2) El contenido del archivo destino (argv[2]) es siempre el mismo o depende de la secuencia de ejecución de los procesos?