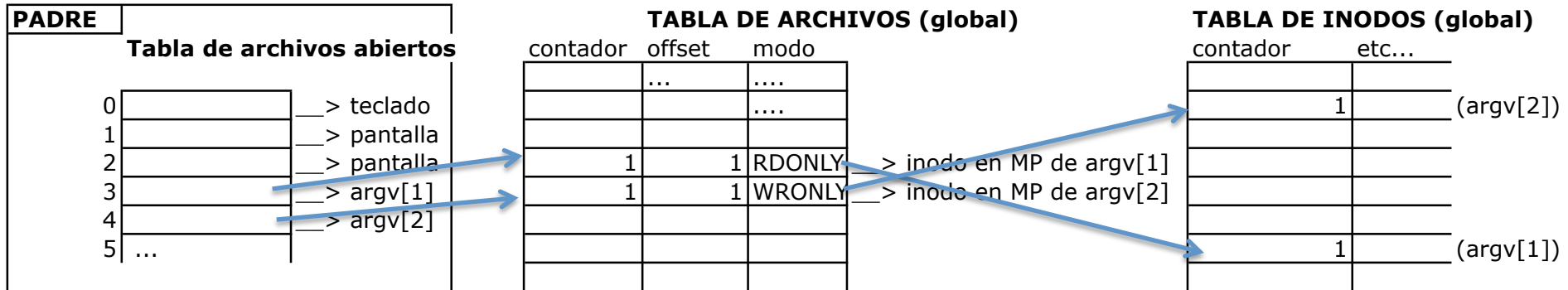


EJEMPLO DE TRATAMIENTO DE ARCHIVOS EN SISTEMAS UNIX

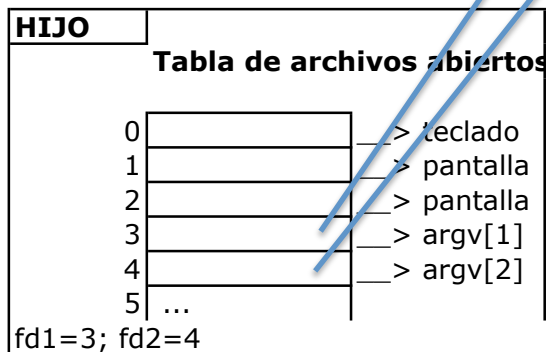
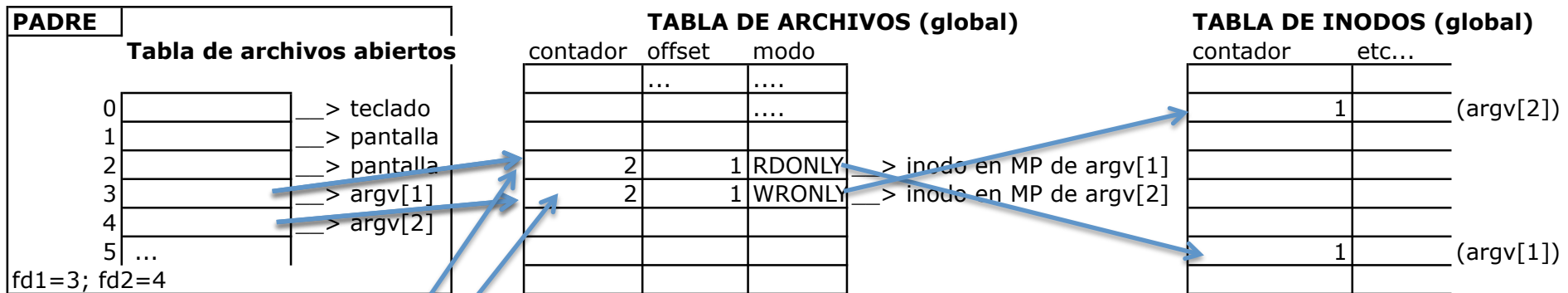
VERSION PRIMERA: LAS APERTURAS ESTAN ANTES DE FORK

Antes de fork:



Tras fork:

OFFSET: PUNTERO DE LECTURA/ESCRITURA

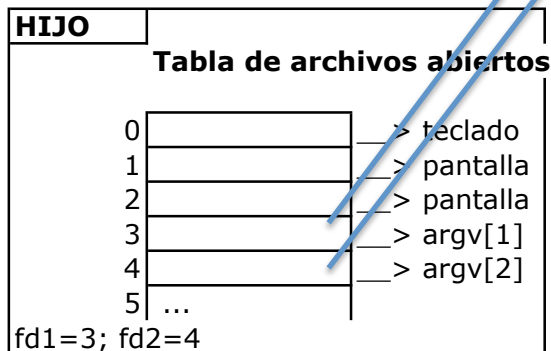
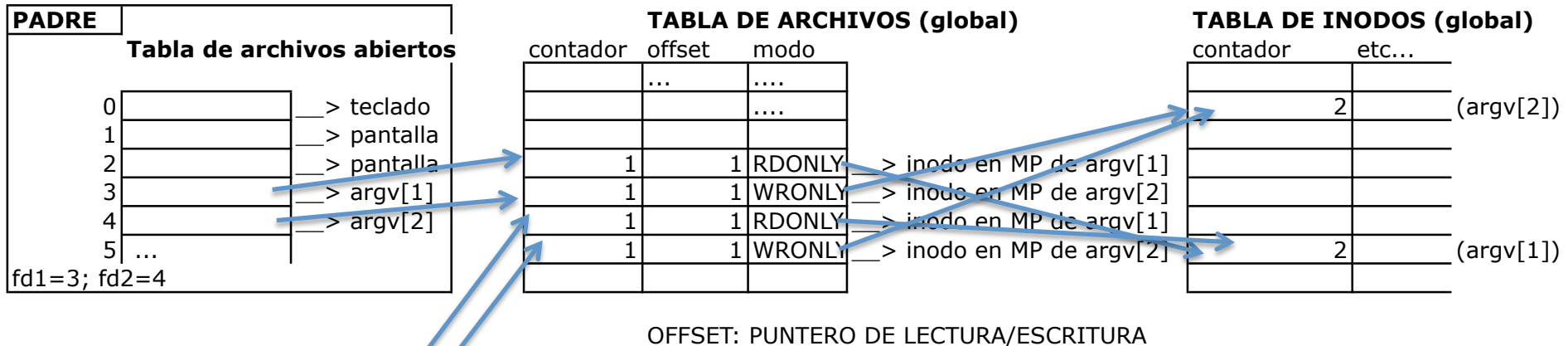


PROGRAMA version primera:

```
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); }
```

VERSION SEGUNDA: LAS APERTURAS ESTAN TRAS FORK

Tras las dos aperturas en los dos procesos:



PROGRAMA versión segunda:

```
char c[2]="\0\0";
fork();
fd1=open(argv[1], O_RDONLY)<0)
.....
fd2=open(argv[2],O_CREAT|O_TRUNC|O_WRONLY)
.....
while (read(fd1,c,1)==1) {
    sleep( random()%4 /10);
    write ( fd2, c,1); }
```

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?