

Programmation systèmes avancée

Wieslaw Zielonka

zielonka@irif.fr

Duplication de descripteurs

duplication de descripteurs

```
./mon_prog <in.txt >out.txt
```

```
cat in.txt | ./mon_prog > out.txt
```

Le programme *mon_prog* lit l'entrée standard (le descripteur 0) et écrit à la sortie standard (descripteur 1).

On demande que le processus exécutant le programme *mon_prog* lise à partir de fichier *in.txt* et qu'il écrive dans *out.txt* **sans qu'on change le code source du programme.**

Solution : dupliquer les descripteurs.

dup()

```
#include <unistd.h>
```

```
int dup(int oldfd);
```

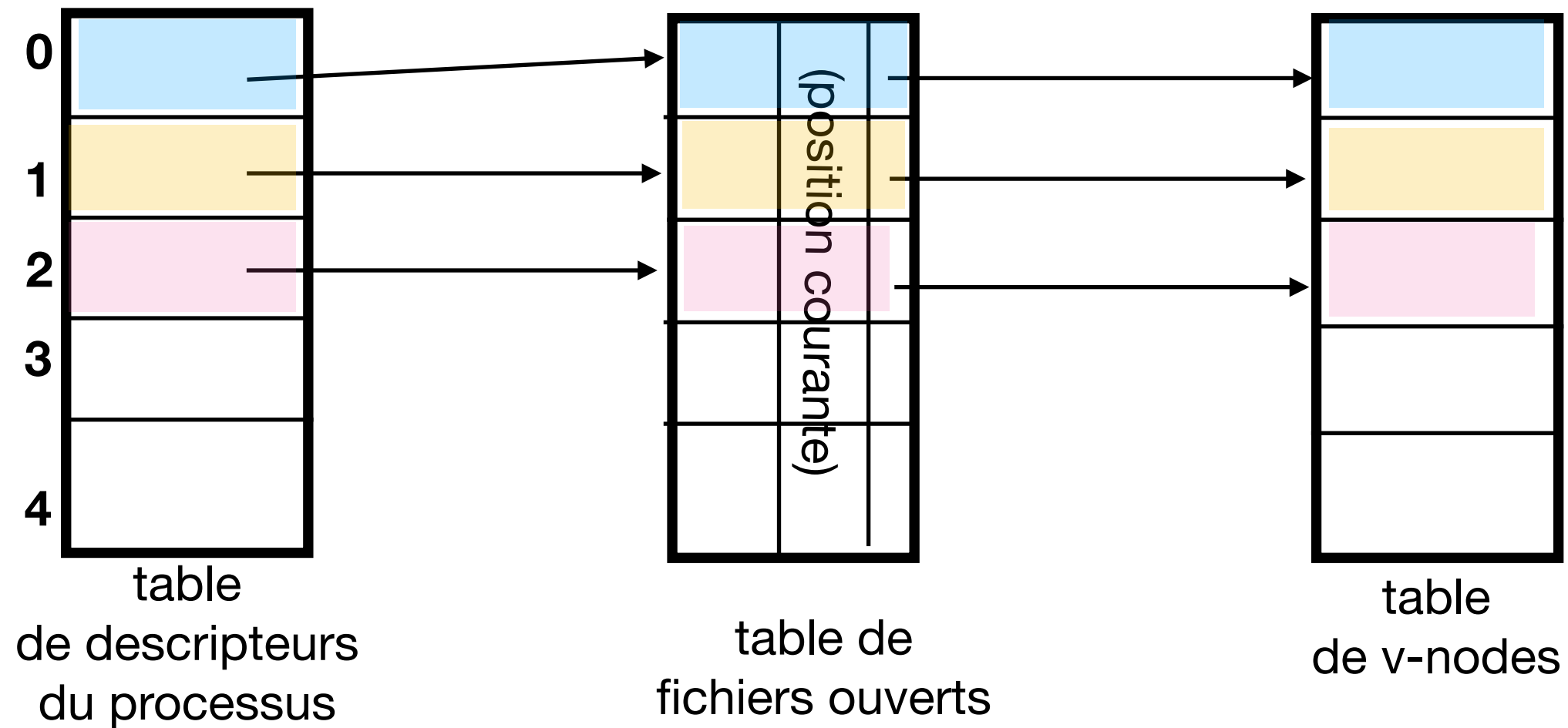
oldfd – un descripteur valide

RETOUR : nouveau descripteur si OK, -1 si erreur

La fonction retourne un nouveau descripteur de fichier qui pointe **vers le même élément de tableau de fichiers ouverts** que le descripteur *oldfd*.

Le système garantit que le descripteur retourné par `dup()` est le plus petit descripteur libre de la table de descripteurs.

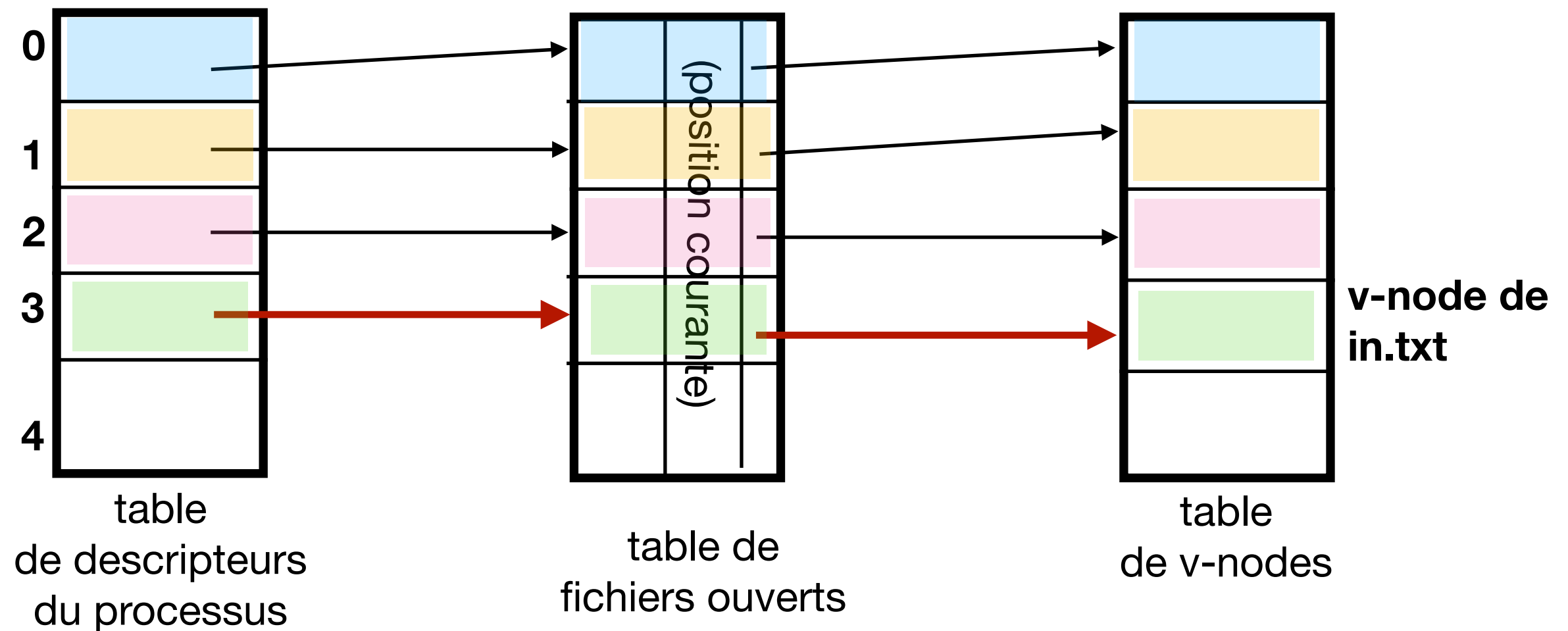
exemple



Configuration initiale, les descripteurs 0, 1, 2 ouverts sur l'entrée standard, sortie standard et sortie d'erreurs standard.

exemple

```
int d = open("in.txt", O_RDONLY);
```

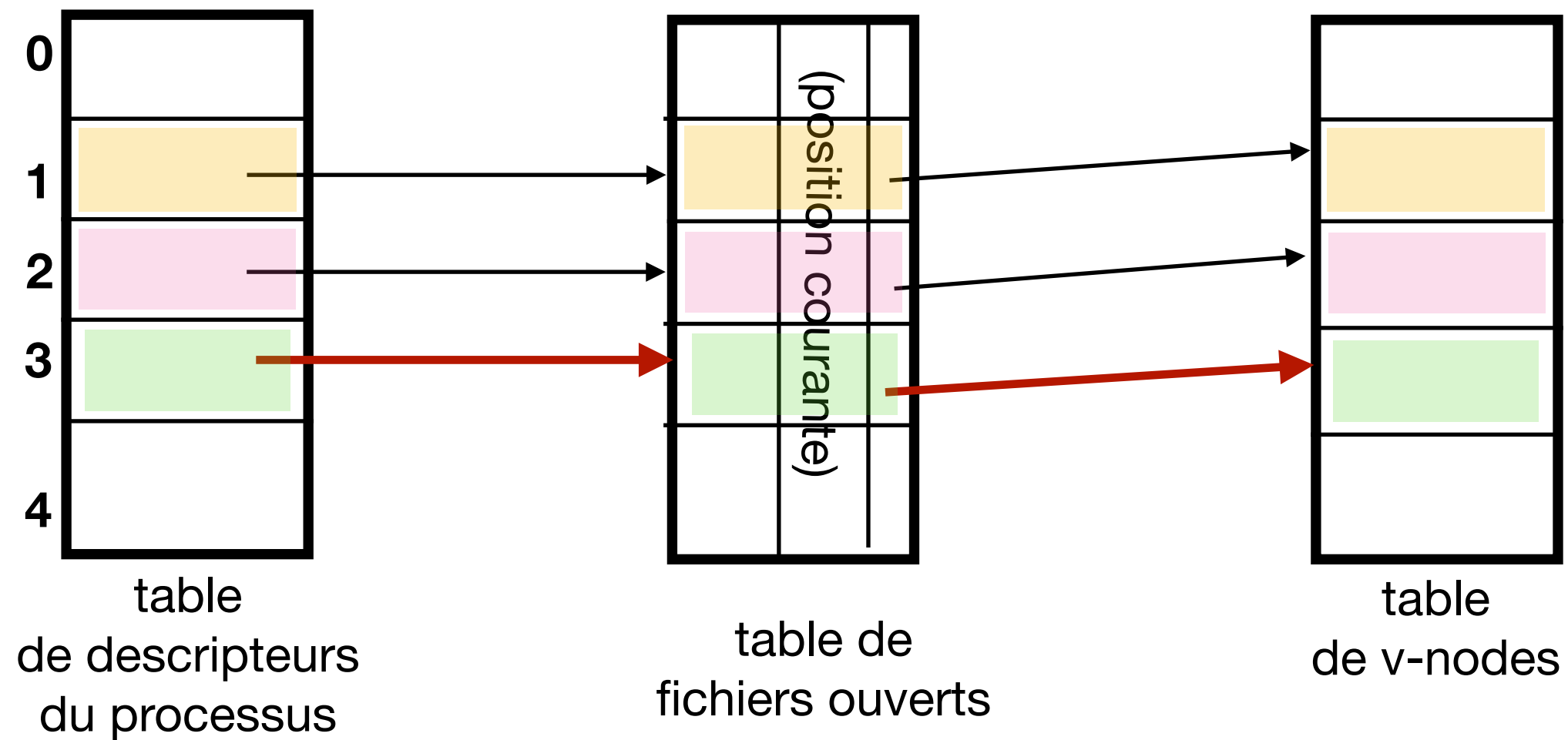


d == 3, nouveau descripteur ouvert

exemple

```
close( 0 );
```

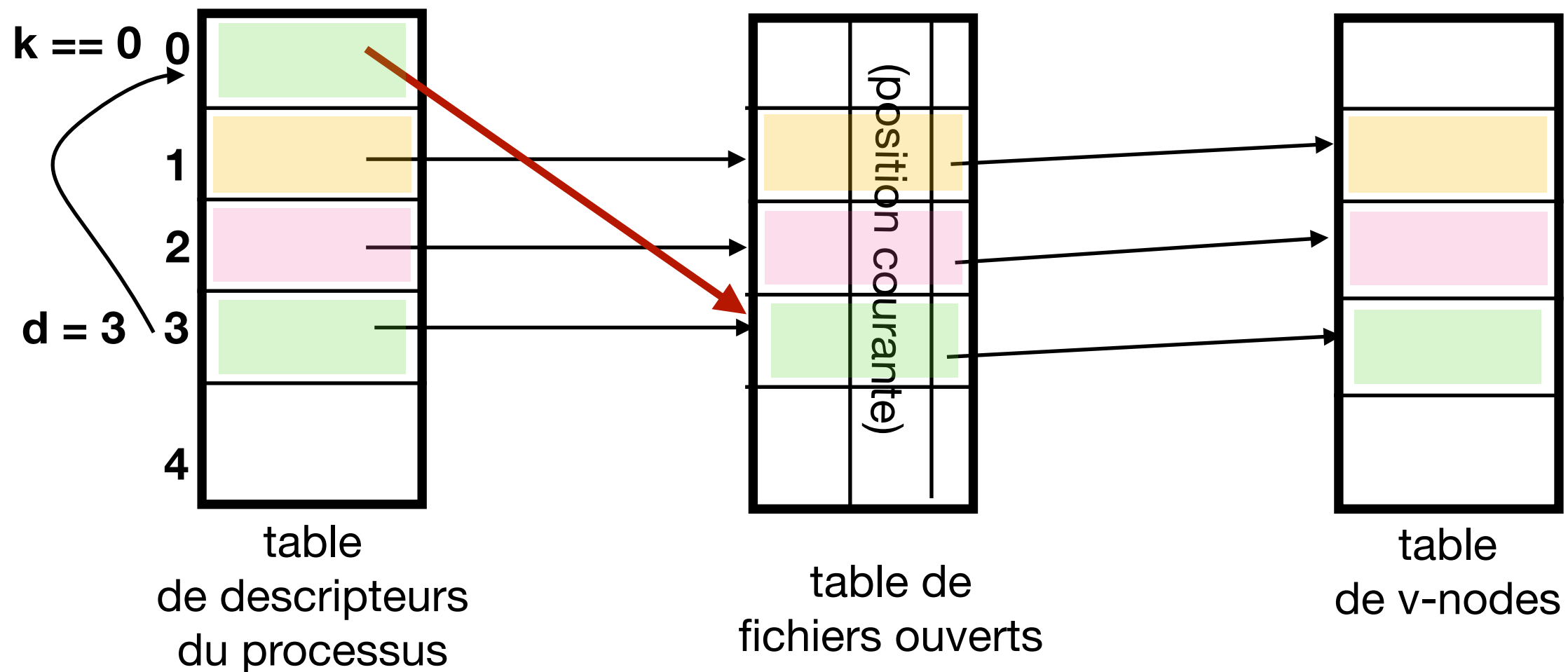
Descripteur 0 devient libre.



exemple dup()

```
int k = dup( d );
```

```
/* d == 3, k <-- 0 */
```

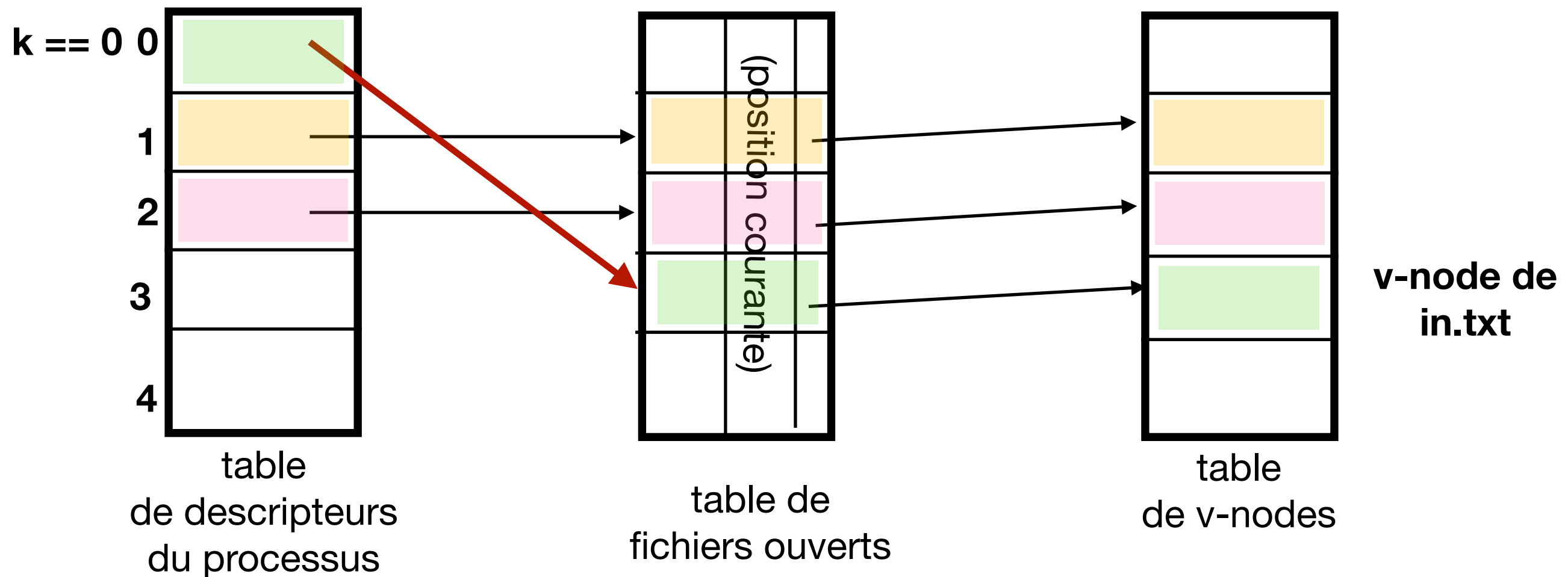


Le descripteur 3 dupliqué dans 0.

exemple

```
close( d );
```

```
/* fermer le descripteur non utilisé */
```



Les `read(0, ...)` sur le descripteur 0 entraîne la lecture de fichier `in.txt`.

dup()

Inconvénient de dup() :

il faut être sûr que le descripteur "cible" soit le premier descripteur libre.

dup2()

```
#include <unistd.h>
```

```
int dup2(int oldfd, int newfd);
```

RETOUR: le nouveau descripteur si OK, -1 si erreur

(si oldfd n'est pas un descripteur valide alors

errno == EBADF)

- si le descripteur newfd est ouvert avant l'appel alors il sera d'abord fermé.
- ensuite le descripteur oldfd est recopié dans newfd

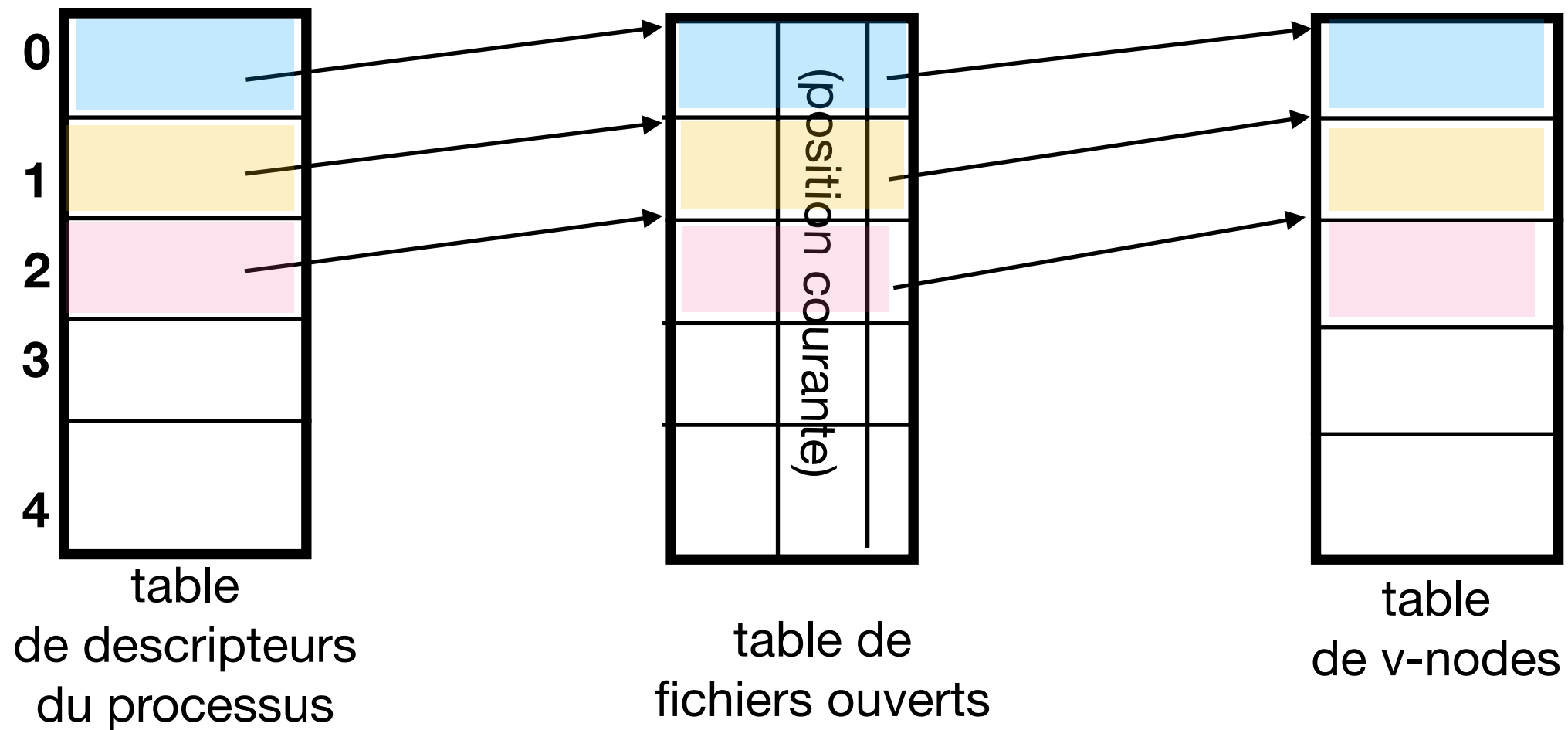
exemple

Redirection de descripteur 1 et 2 dans le fichier `out.txt` :

```
int d = open("out.txt", O_WRONLY);  
dup2(d, 1);  
dup2(d, 2);  
/* maintenant les descripteurs 0, 1, d  
 * pointent vers le même fichier out.txt ouvert  
 * en écriture*/  
close( d );
```

l'écriture dans les descripteurs 1 et dans 2 entraînera l'écriture dans le fichier `out.txt`.

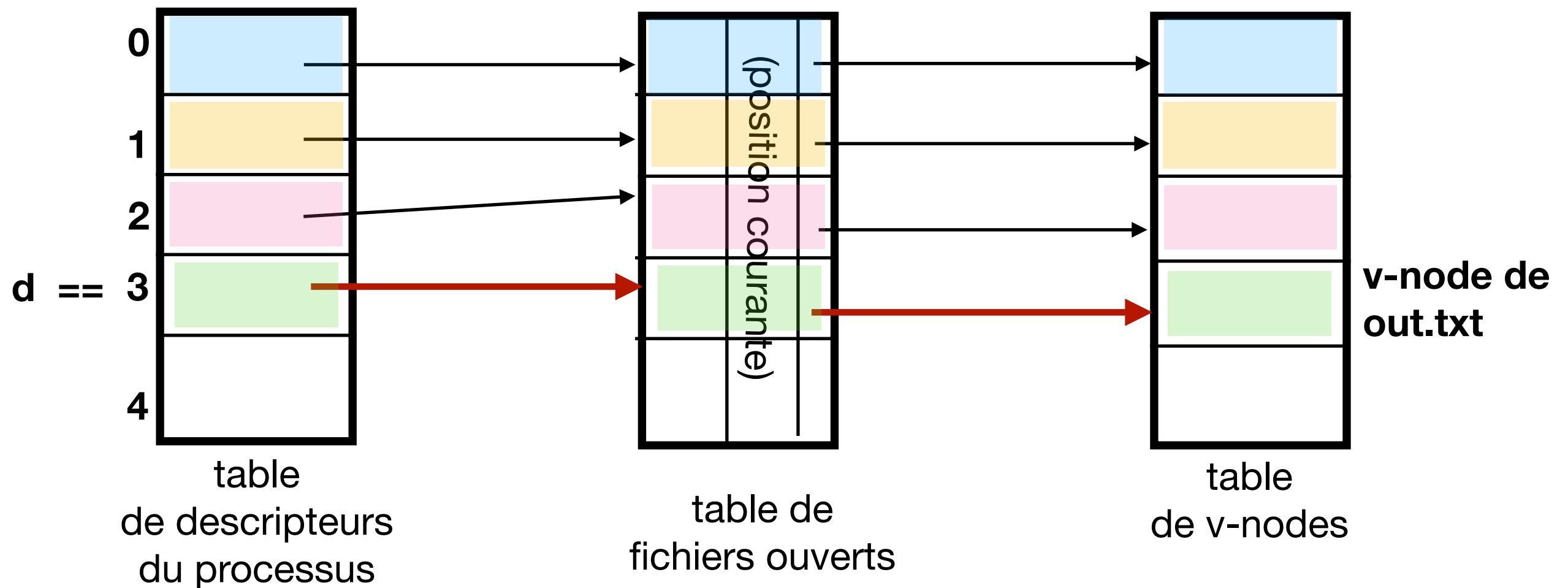
exemple



Configuration initiale, les descripteurs 0, 1, 2 ouverts sur l'entrée standard, sortie standard et sortie d'erreurs standard.

exemple

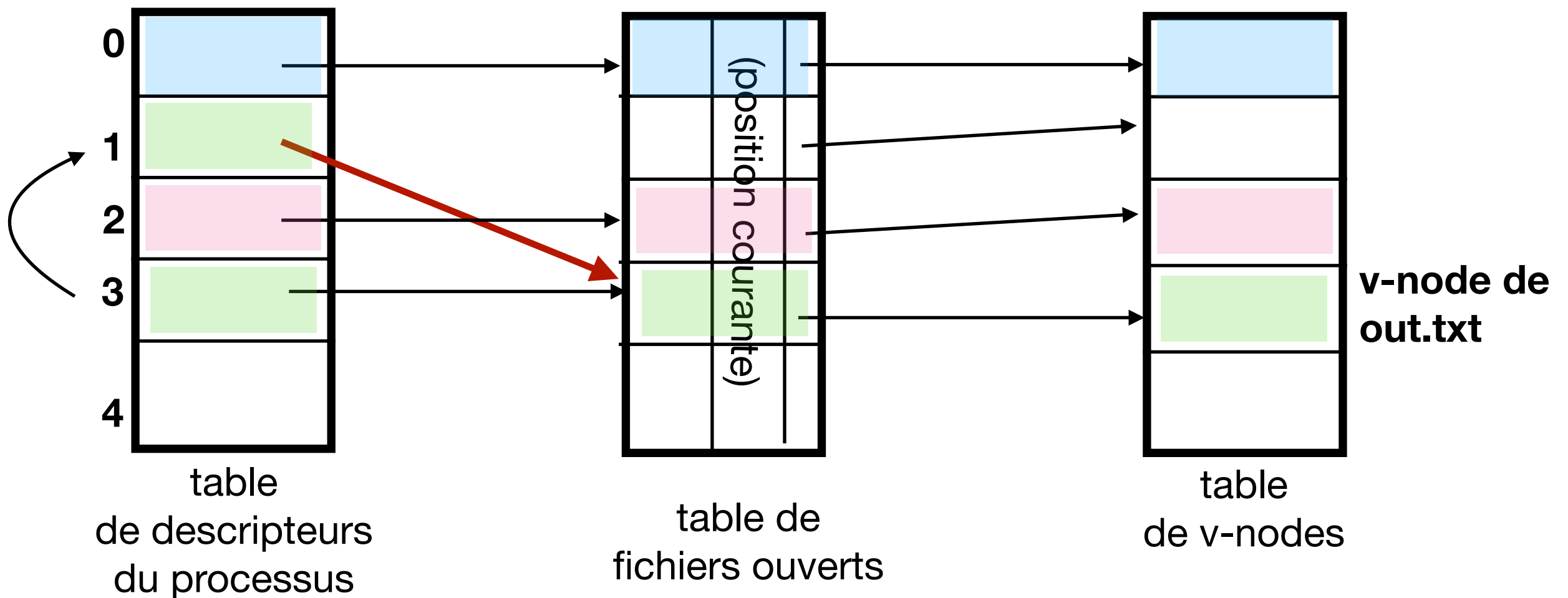
```
int d = open("out.txt", O_WRONLY);
```



d == 3, nouveau descripteur ouvert

exemple

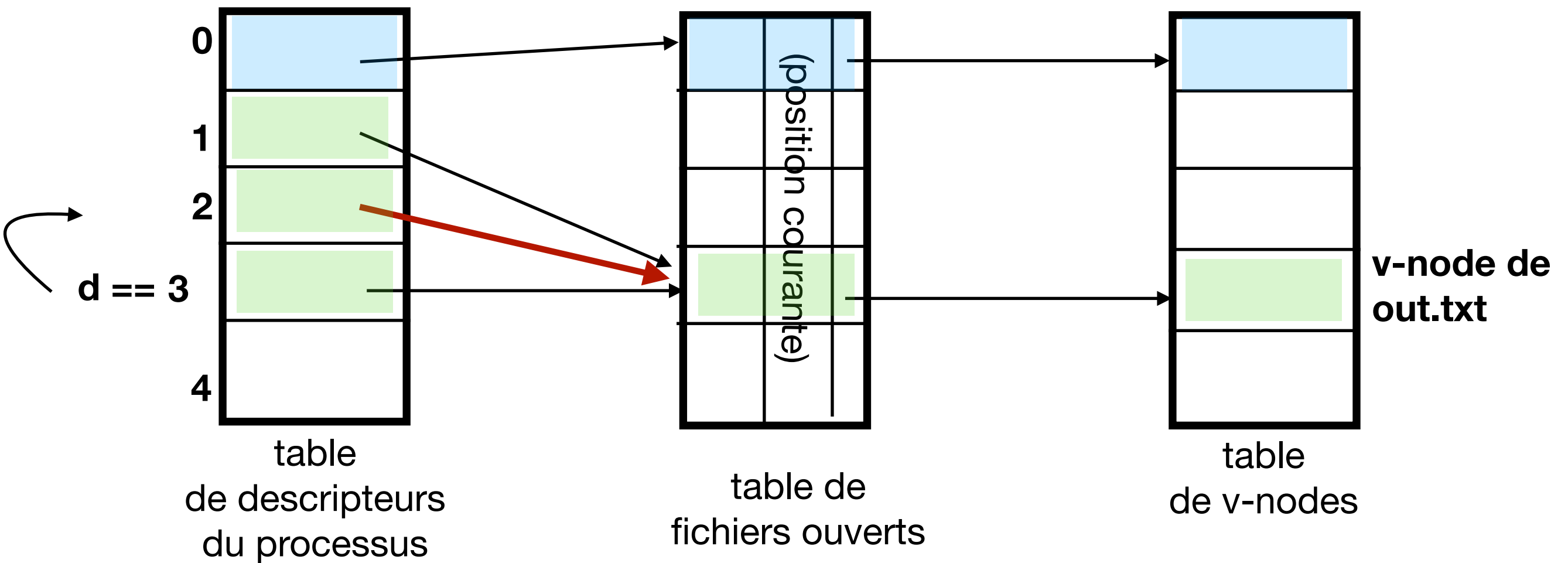
```
dup2( d, 1 );
```



descripteur 1 ferme
descripteur 3 (d == 3) recopié dans 1

exemple

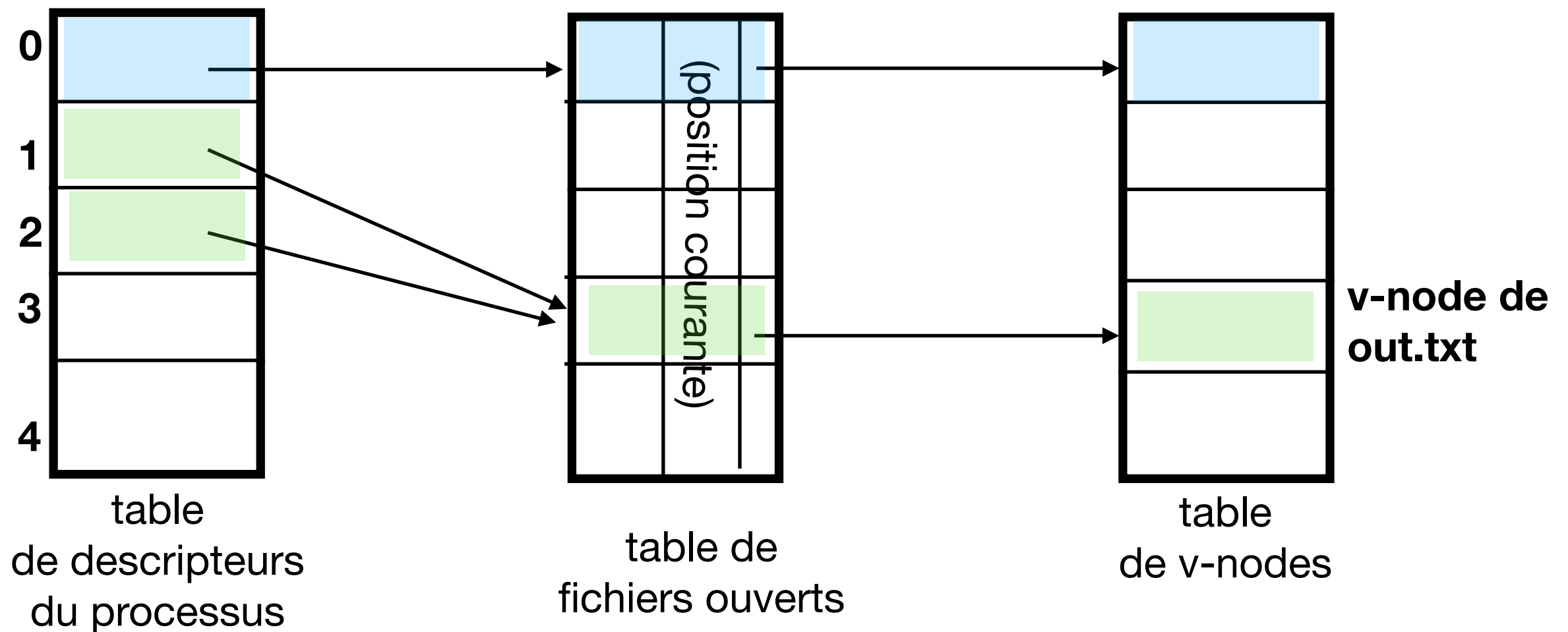
```
dup2( d, 2 );
```



descripteur 2 ferme
descripteur 3 (`d == 3`) recopié dans 2

exemple

```
close( d );
```



Maintenant les écritures sur les descripteurs 1 et 2 entraîneront l'écriture dans le fichier out.txt