

Sistemas Operativos

2018/2019

Dup

- Cria uma cópia de um descritor de ficheiro

Processo

TP

TF

v-node/i-node (simplificado)

1123

0	
1	
2	
3	
4	

modo	w
pos	8
#ref	2
vptr	

modo	r
pos	0
#ref	1
vptr	

filename	out.txt
size	16
#ref	1
...	

filename	foo.txt
size	256
#ref	2
...	

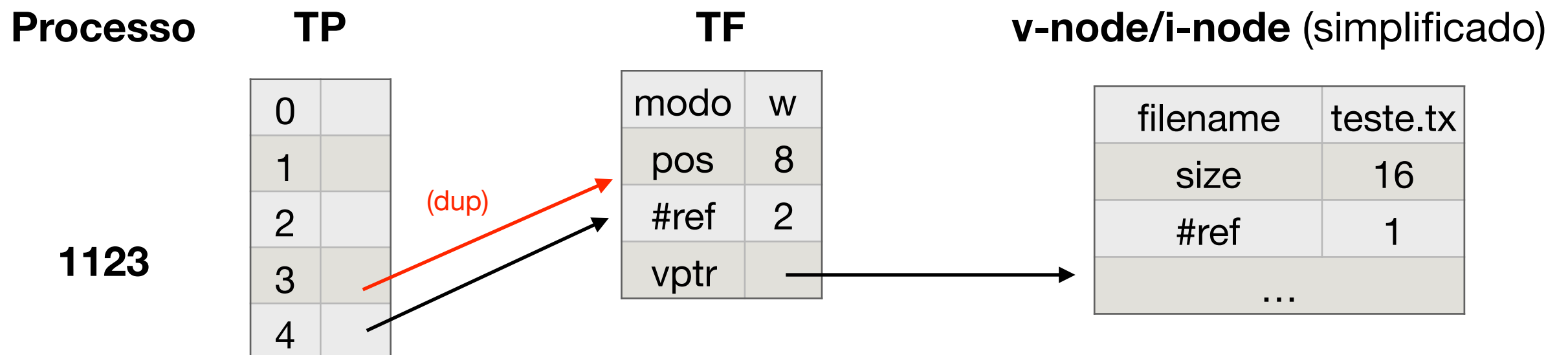
(dup)

Chamadas ao sistema

- Bibliotecas
 - `<unistd.h>` - definições e declarações de chamadas

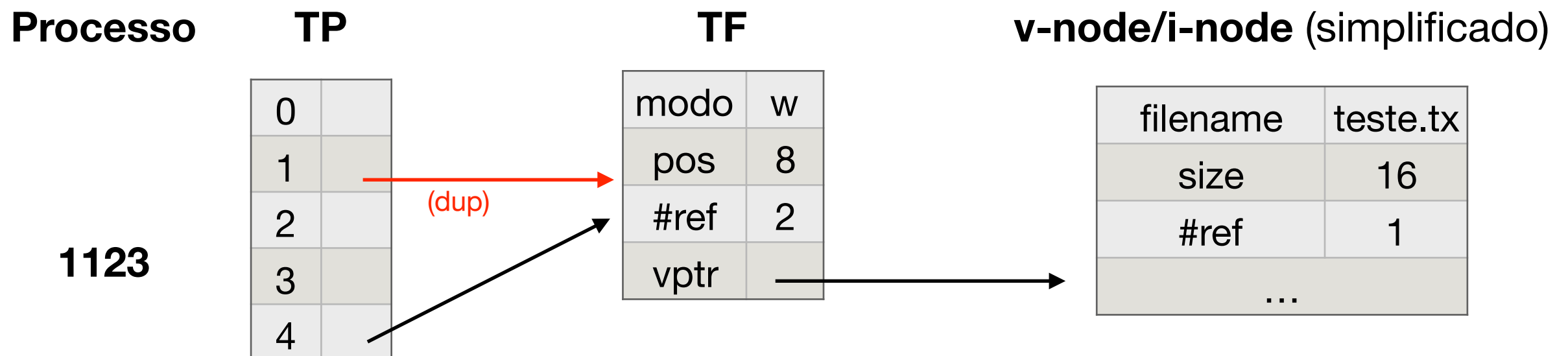
Chamadas ao sistema

- `int dup(int fd);`
 - cria uma cópia do descritor **fd**
 - devolve cópia do descritor ou erro (p.ex: se o descritor fd não é válido)
 - escolhe o descritor disponível com o menor número
 - mantém modo e posição do descritor original



Chamadas ao sistema

- `int dup2(int fd1, int fd2);`
 - cria uma cópia do descritor **fd1** no descritor **fd2**, especificado pelo utilizador
 - devolve valor de fd2 ou erro
 - se o descritor fd2 estiver aberto é chamado um **close()** implícito
 - mantém modo e posição do descritor original
 - No exemplo em baixo *dup2(4,1)* qual o comportamento de um `printf()`?



Considerações

- O descritor a ser copiado deve ser fechado (função *close()*) depois de ser realizado o comando *dup()* caso já não seja necessário.
- Ao redirecionar o *stdout* para um ficheiro, o tamanho do buffer de memória passa de uma linha para o tamanho utilizado pelo sistema de ficheiros (tipicamente 4KB). Funções como a *printf()* deixam por isso de fazer buffering à linha.
 - A função *fflush(stdout)* permite forçar a persistência dos dados em buffer
 - A função *setbuf(stdout, NULL)* permite desativar o buffer de memória

Material de Apoio

- <https://www.usna.edu/Users/cs/aviv/classes/ic221/s16/lec/21/lec.html>