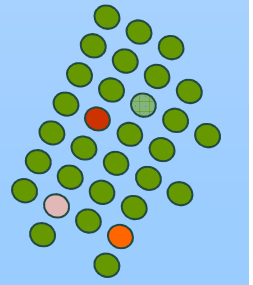
I/O non bufferizzato (2)

Capitolo 3 -- Stevens



Efficienza di I/O

```
#include "ourhdr.h"

#define BUFFSIZE    8192

int main(void)
{
   int   n;
   char buf[BUFFSIZE];

   while((n = read(STDIN_FILENO,buf,BUFFSIZE))>0)
       if (write(STDOUT_FILENO,buf, n) != n)
              err_sys("write error");
  if (n < 0)   err_sys("read error");
   exit(0);
}
```

-apertura file standard

-chiusura file

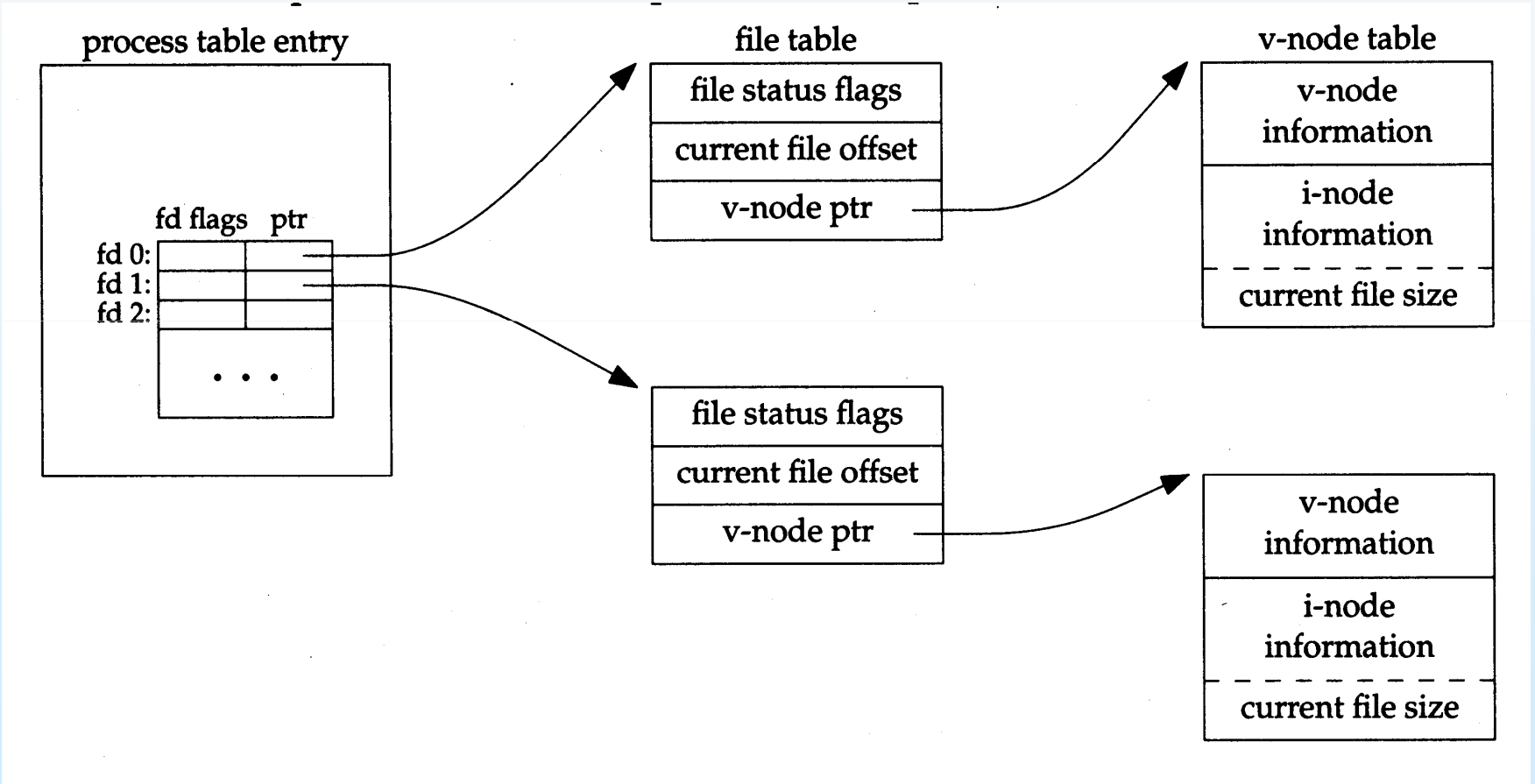
- scelta di BUFFERSIZE

Condivisione di file

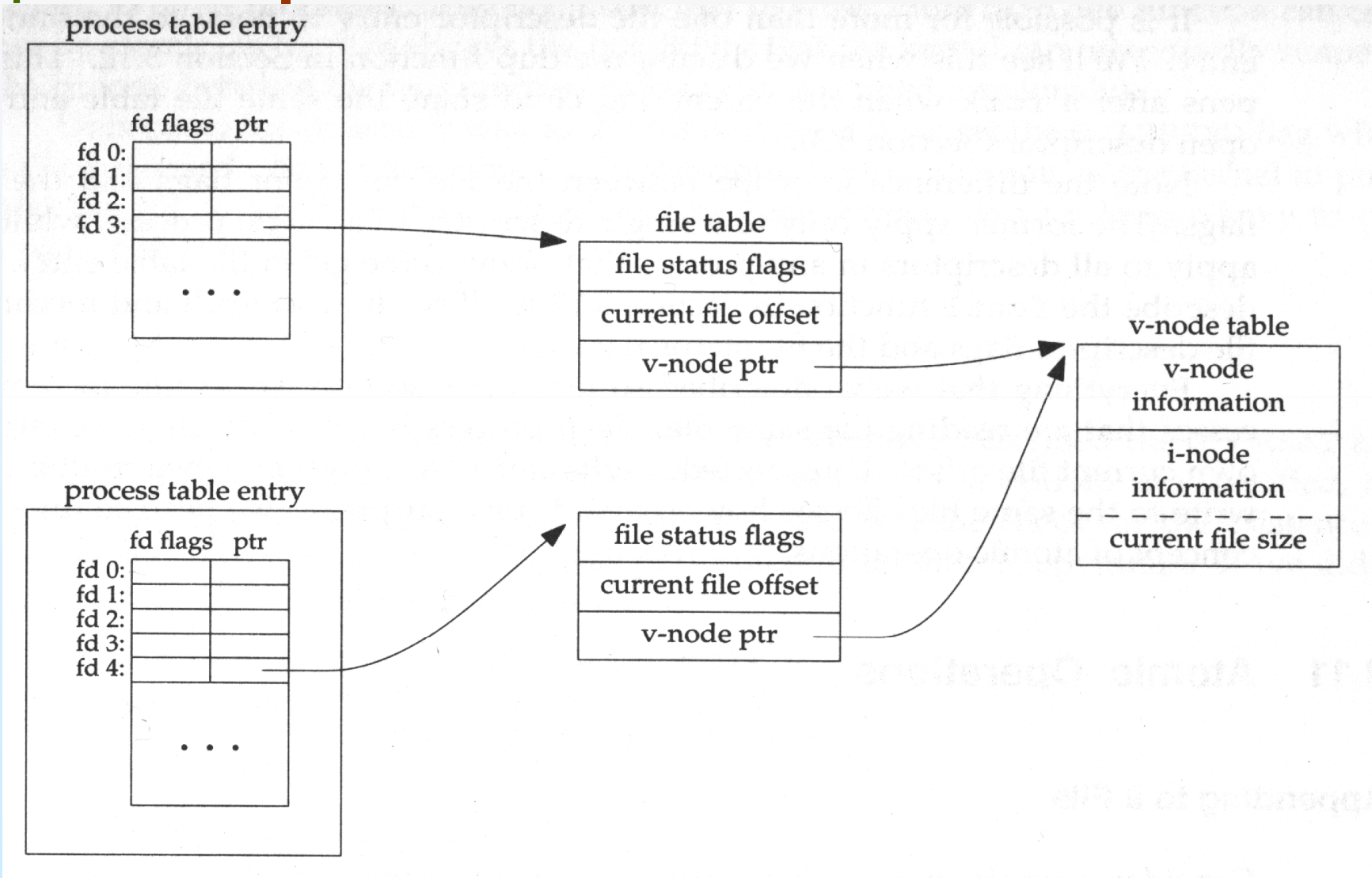
- Unix supporta la possibilità che più processi condividano file aperti

- Prima di analizzare questa situazione esaminiamo le strutture dati che il kernel utilizza per I/O
  - 3 strutture dati per l'I/O

Strutture dati di file aperti



2 processi su uno stesso file



Operazioni Atomiche

- Immaginate il seguente scenario:
  - 2 processi aprono lo stesso file
  - ognuno si posiziona alla fine e scrive (in 2 passi)
    1. lseek(fd, 0 ,SEEK_END);
    2. write (fd, buff , 100);
  - se il kernel alterna le due operazioni di ogni processo si hanno

effetti indesiderati

Operazioni Atomiche

- Unix risolve il problema:
  - Apre il file con il flag "O_APPEND"
  - Questo fa posizionare l'offset alla fine, prima di ogni write
  - In altre parole, le operazioni di
    1. posizionamento
    2. write

    } sono atomiche

In generale una operazione atomica è composta da molti passi che o sono eseguiti tutti insieme o non ne è eseguito nessuno

dup & dup2

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

int dup(int *filedes*);

int dup2(int *filedes*, int *filedes2*);

Descrizione: assegnano un altro fd ad un file che già ne possedeva uno, cioè *filedes*

Restituiscono entrambe: il nuovo fd se OK

-1 altrimenti

dup & dup2

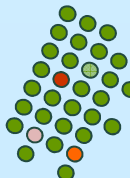
- In particolare:

int dup(int *filedes*);

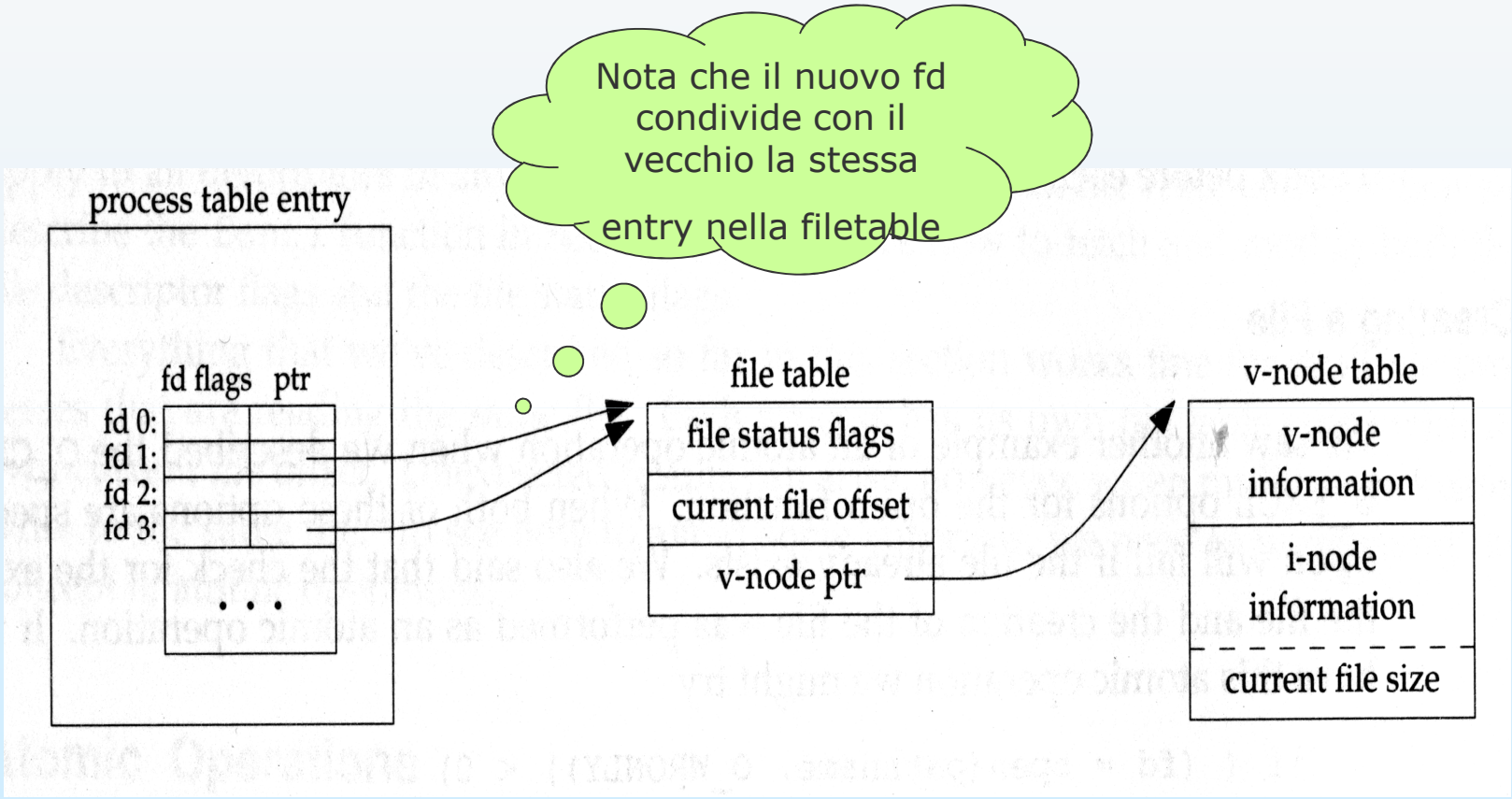
- restituisce il più piccolo fd disponibile

int dup2(int *filedes*, int *filedes2*);

- Assegna al file avente già file descriptor *filedes* anche il file descriptor *filedes2*
  - Se *filedes2* è già open esso è prima chiuso e poi è assegnato a *filedes*
  - Se *filedes2=filedes* viene restituito direttamente *filedes2*
- dup2 è una operazione atomica



DS del kernel, dopo dup



esercizi

1. copiare un file in un altro usando solo le funzioni di standard I/O *getchar* e *putchar*
   - hint: "duplicare" gli standard file

2. copiare il contenuto di un file in un altro usando esclusivamente read da standard input e write su standard output