

Questão 7

Variáveis globais

```
char * filename = "...";
```

```
char vector[SIZE]; // vector para os bytes lidos
```

```
int primeira_vez = 1;
```

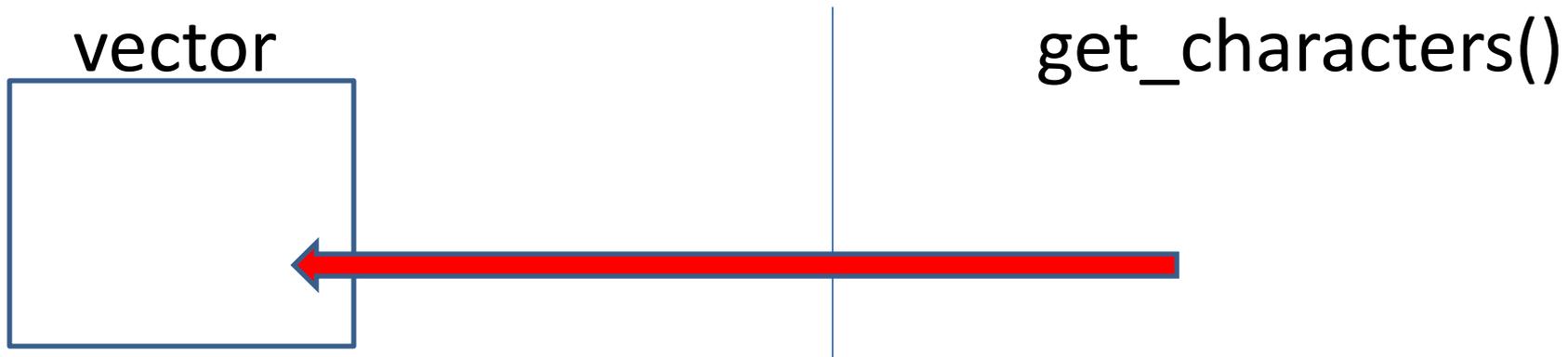
```
char B[BLOCK];
```

```
int ocupados = 0; // número de bytes no buffer B
```

```
int fd; // número de canal para ficheiro
```

```
int regEOF = 0; // encontra fim de ficheiro ao ler...
```

```
int lidos;
```

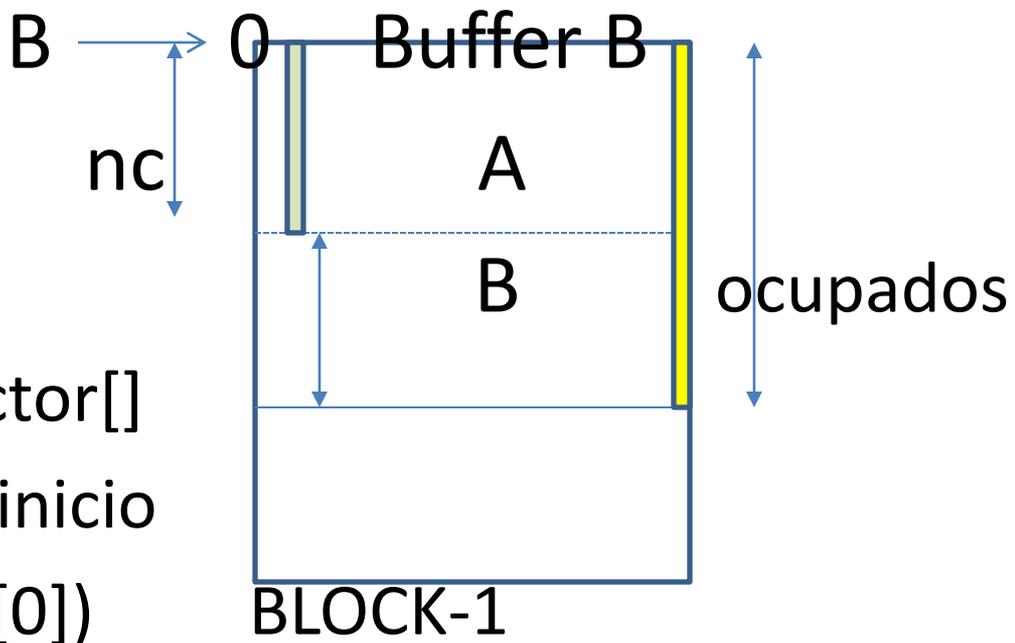


```
int get_characters (char * destin, int nc);
{ int copiados; // número de chars copiados

if (primeira_vez) {
    fd = open(filename, O_RDONLY);
        // ignora erros
    primeira_vez = 0;
}

if (nc < ocupados) { // há chars suficientes em B
.....
```

```
{ copiados = nc;
```



```
... copiar zona A para vector[]  
... transferir zona B para inicio  
de buffer B (posicao B[0])
```

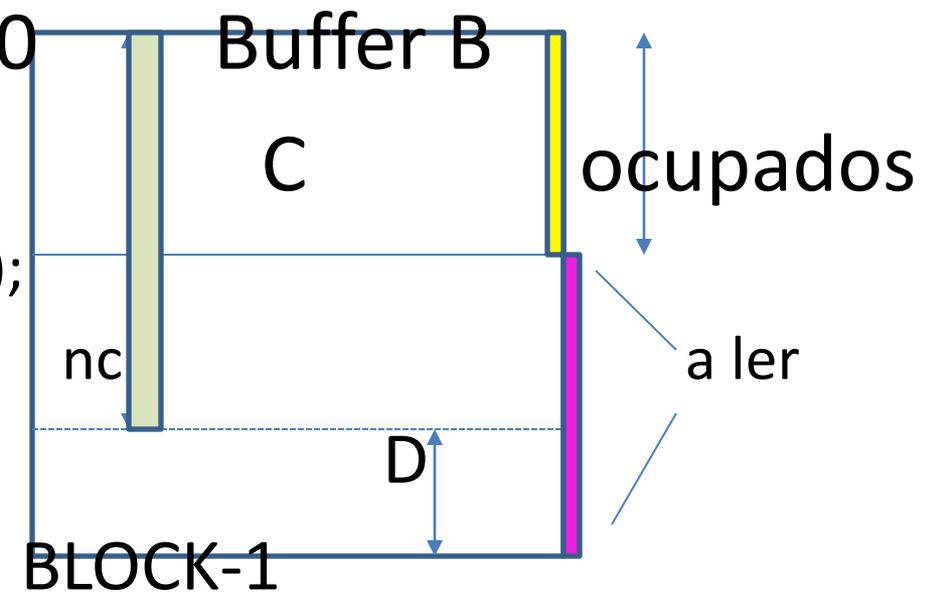
```
ocupados = ocupados - copiados;  
return copiados;  
} // caso em que nc < ocupados
```

else { // nc >= ocupados -- não há chars suficientes

```

lidos = read(fd, B+ocupados,
            BLOCK-ocupados);
if (lidos == 0) regEOF = 1;
ocupados = ocupados + lidos;
copiados = min(nc, ocupados);

```

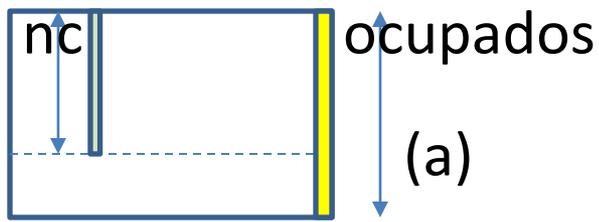


```

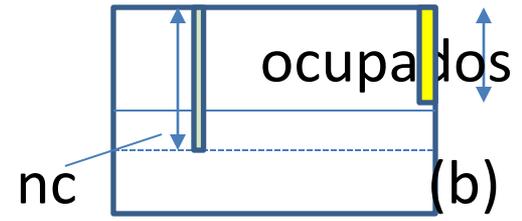
... copia copiados para vector[] -- ver casos (a) e (b)
... transferir D para início de B, se houver chars
ocupados = ocupados - copiados;
return copiados;

```

}



DEPOIS de LER



Questão 16

a) A execução do programa produzirá uma sucessão de linhas no canal standard out (ligado ao écran), que apresentarão de forma entrelaçada, as cadeias enviadas por cada chamada ao SO `write(1,...)` por cada um dos processos. Cada processo originará 5 linhas: no caso do Pai, contendo a cadeia "Pai", no caso do Filho, contendo a cadeia "Filho".

Em cada ponto da execução, a explicação das acções depende directamente da semântica das chamadas ao SO `fork()` e `write()`. Consulte os elementos de apoio na página da disciplina.

b)

```
int fd;
```

```
char *name[MAX];
```

```
if (fork() == 0) {
```

```
    strcpy(buf, "Filho\n");
```

```
    sprintf(name, "/tmp/filho_%d", getpid());
```

```
    /*afecta name com string "temp/filho_FPID" */
```

```
/*esta acção poderia ser indicada em pseudo-código
```

```
desde que indicasse a acção de concatenação
```

```
e recorresse à chamada ao SO getpid()*/
```

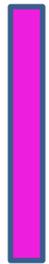
```
    } else {
```

```
        strcpy(buf, "Pai\n");
```

```
        sprintf(name, "/tmp/pai_%d", getpid());
```

```
        /*afecta name com string "temp/pai_PPID" */
```

```
    }
```



```
fd = open(name, O_WRONLY);
```

```
...
```

```
write(fd, ...)
```