



Problema Produtor/Consumidor

1 Informações Gerais

Podemos enunciar o trabalho da seguinte forma: dois processos compartilham uma mesma área de memória (no nosso caso compartilham o mesmo arquivo). Um processo irá escrever ao final do arquivo um valor aleatório $r \in [0, 99]$. Este processo é chamado produtor. O outro processo, o consumidor, irá remover estes valores do início do arquivo (imprimindo-o na tela). O produtor irá inserir um valor no arquivo e irá esperar s segundos, onde s é um inteiro aleatório $s \in [1, 3]$. O consumidor também usará a mesma lógica de espera para retirar elementos do arquivo. O nome do arquivo será `buffer.txt` e será inicializado com 10 inteiros (um inteiro por linha). Para não haver problemas de concorrência ao acessar o arquivo `buffer.txt`, antes de escrever ou ler o arquivo `buffer.txt` os processos devem criar um arquivo `buffer.txt.lock` que indica que o `buffer.txt` está sendo utilizado. Ao terminar a escrita ou leitura deve-se apagar o `buffer.txt.lock`. Assim, antes de escrever ou ler um arquivo deve-se verificar se existe o arquivo `buffer.txt.lock`. Se esse arquivo existe então o processo deverá esperar até que esteja disponível ler ou escrever no `buffer.txt`. O trabalho deverá ser feito usando a chamada `fork`. Exemplo de execução:

```
$ ./produtor-consumidor
[Consumidor] 91
[Consumidor] 10
[Produtor] 62
[Consumidor] 33
[Consumidor] 79
[Produtor] 9
[Produtor] 52
[Produtor] 34
```