

# Trabalho 1 - Sistemas Operacionais

## Relatório

**Daiane Focking Andrade<sup>1</sup>**

<sup>1</sup>Ciência da Computação  
Centro de Desenvolvimento Tecnológico  
Universidade Federal de Pelotas (UFPel)

dfandrade@inf.ufpel.edu.br

**Abstract.** *Report about Producer-Consumer problem in C developed in Sistemas Operacionais 2016/2 class using [Barney 2016] e [SBC 2016] as standard.*

**Resumo.** *Relatório descritivo de atividades sobre o problema Producer-Consumer feito em C para a disciplina de Sistemas Operacionais 2016/2, utilizando [Barney 2016] e [SBC 2016] como base.*

### Descrição de Hardware

Samsung Notebook ATIV Book 4  
Intel Core i5-3230M 2.60GHz  
4GB RAM DDR3  
500GB HD S-ATA II  
OS Ubuntu 16.04 LTS Xenial Xerus

### Casos de teste

Número de repetições: 20.

Número de casos de teste: 4.

Quantidade de linhas por caso de teste: 50, 5.000, 25.000, 50.000.

Todos os arquivos tinham em cada linha a seguinte `string`: “*hello world hello*”, mas também foi testado com arquivos que tinham diferentes estruturas de texto.

- **Teste 1:** 50 linhas, resultado esperado 100 ocorrências;
- **Teste 2:** 5.000 linhas, resultado esperado 10.000 ocorrências;
- **Teste 3:** 25.000 linhas, resultado esperado 50.000 ocorrências;
- **Teste 4:** 50.000 linhas, resultado esperado 100.000 ocorrências;

### Como executar

- **Versão Sequencial:** `make testel`
- **Versão Paralela:** `make testelt`.

## Resultados

O tempo de execução foi medido através da seguinte forma:

```
clock_t begin = clock();
\*execução de ocorrencia()*\
clock_t end = clock();
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```

Todos dos tempos apresentado estão em segundos.

Sequencial				
	Teste 1	Teste 2	Teste 3	Teste 4
Média	0,00001225	0,00041415	0,00184605	0,0044372

Paralela				
	Teste 1	Teste 2	Teste 3	Teste 4
Média	0,00114735	0,03188945	0,13525095	0,2535692

SpeedUp				
	Teste 1	Teste 2	Teste 3	Teste 4
Média	0,01329131176	0,01326660767	0,01374625793	0,01749518852

O conjunto completo de dados está anexado ao final do arquivo.

## Análise

Como o esperado, a versão sequencial apresentou em média um tempo 1,44% melhor que a versão paralela, uma vez que não foram implementados `locks` individuais por posição. É possível que se isso fosse feito, não houvesse esse comportamento. Mesmo quando testado com buffers de diferentes tamanhos de buffer, o comportamento era semelhante.

## Anexo: Tabelas de Execução

## References

Barney, B. (2016). Posix threads programming. <https://computing.llnl.gov/tutorials/pthreads/>. Acessado em Setembro de 2016.

SBC (2016). Modelos para publicação de artigos. <http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>. Acessado em Setembro de 2016.

**Table 1. Execução Sequencial**

Sequencial				
Execução n°	Teste 1	Teste 2	Teste 3	Teste 4
1	0,000024	0,000572	0,002524	0,005053
2	0,000011	0,000539	0,002596	0,005453
3	0,00001	0,000515	0,002578	0,005388
4	0,000009	0,000527	0,002637	0,004673
5	0,000009	0,000523	0,002052	0,00488
6	0,00001	0,000535	0,001643	0,004758
7	0,000009	0,000381	0,001653	0,004524
8	0,000009	0,000352	0,001609	0,004438
9	0,000051	0,000349	0,001578	0,004403
10	0,00001	0,000386	0,001609	0,004206
11	0,00001	0,000359	0,001574	0,004278
12	0,000009	0,00035	0,001611	0,004221
13	0,000009	0,000351	0,001659	0,004035
14	0,000009	0,000357	0,001647	0,004027
15	0,000009	0,000356	0,001895	0,004087
16	0,000009	0,000357	0,001656	0,003884
17	0,000009	0,000361	0,001611	0,003914
18	0,000009	0,000389	0,001605	0,004827
19	0,00001	0,000371	0,001607	0,00395
20	0,00001	0,000353	0,001577	0,003745
Média	0,00001225	0,00041415	0,00184605	0,0044372

**Table 2. Execução Paralela**

Paralelo				
Execução n°	Teste 1	Teste 2	Teste 3	Teste 4
1	0,000986	0,054421	0,241083	0,266673
2	0,001069	0,047981	0,178223	0,252076
3	0,00168	0,044664	0,145208	0,254
4	0,000908	0,035315	0,127447	0,257523
5	0,000845	0,03919	0,130308	0,249338
6	0,001	0,036897	0,128506	0,252847
7	0,001104	0,0332	0,126753	0,251014
8	0,001295	0,031795	0,128241	0,254598
9	0,000788	0,031728	0,130912	0,254914
10	0,001484	0,029716	0,130144	0,251864
11	0,001132	0,026552	0,127145	0,251541
12	0,000849	0,024285	0,12554	0,256757
13	0,001469	0,026596	0,127225	0,247449
14	0,000869	0,023577	0,126188	0,250886
15	0,002607	0,024324	0,12609	0,254123
16	0,002293	0,025995	0,115096	0,25597
17	0,000961	0,025563	0,117625	0,25313
18	0,000462	0,024558	0,124836	0,253925
19	0,000672	0,025922	0,123467	0,250141
20	0,000474	0,02551	0,124982	0,252615
Média	0,00114735	0,03188945	0,13525095	0,2535692

**Table 3. SpeedUp**

SpeedUp				
Execução nº	Teste 1	Teste 2	Teste 3	Teste 4
1	0,02434077079	0,01051064846	0,01046942339	0,01894829998
2	0,01028999065	0,01123361331	0,01456602122	0,02163236484
3	0,005952380952	0,01153053914	0,01775384276	0,02121259843
4	0,009911894273	0,01492283732	0,02069095389	0,01814595201
5	0,01065088757	0,01334524113	0,01574730638	0,0195718262
6	0,01	0,01449982383	0,01278539523	0,01881770399
7	0,008152173913	0,01147590361	0,01304111145	0,01802289912
8	0,00694980695	0,0110709231	0,01254668944	0,01743140166
9	0,06472081218	0,01099974786	0,0120538988	0,0172724919
10	0,006738544474	0,01298963521	0,01236322842	0,01669948861
11	0,008833922261	0,01352063875	0,01237956664	0,01700716782
12	0,01060070671	0,01441218859	0,01283256333	0,01643966864
13	0,006126616746	0,0131974733	0,01303988996	0,01630639041
14	0,01035673188	0,01514187556	0,01305195423	0,01605111485
15	0,003452243959	0,0146357507	0,01502894758	0,01608276307
16	0,003924989097	0,01373341027	0,01438798916	0,01517365316
17	0,009365244537	0,01412197316	0,01369606801	0,01546241062
18	0,01948051948	0,01584005212	0,01285686821	0,01900955006
19	0,01488095238	0,01431216727	0,01301562361	0,01579109382
20	0,02109704641	0,0138377107	0,01261781697	0,01482493122
Média	0,01329131176	0,01326660767	0,01374625793	0,01749518852