

# Relatório do Trabalho de Conclusão da Disciplina



**UFU**

Trabalho de Sistemas Operacionais

Alunos:

Henrique de Moraes Segatto  
Henrique Corrêa de Oliveira  
Gustavo de Jesus Gomes  
Diego Batistuta Ribeiro de Andrade

# Índice

|                                       |    |
|---------------------------------------|----|
| 1-Apresentação do Trabalho.....       | 03 |
| 1.1-Máquina Usada.....                | 03 |
| 1.2-Sistemas Operacionais Usados..... | 03 |
| 1.3-Programa usado para testes.....   | 04 |
| 2-Testes Efetuados.....               | 04 |
| 2.1-Sistema Linux.....                | 04 |
| 2.2-Sistema Solaris.....              | 07 |
| 2.3-Sistema FreeBSD.....              | 12 |
| Observações Adicionais.....           | 16 |
| 3-Conclusão.....                      | 17 |

# 1-Apresentação do trabalho

Neste trabalho pretendemos testar a execução do programa feito na Atividade Prática 3 nos Sistemas Operacionais Linux, Solaris e FreeBSD e comparar a multiprogramação, o multiprocessamento e a Comunicação entre Processos (IPC) de cada SO.

## 1.1-Máquina usada

Notebook HP G42, Intel Core i5 CPU M 460 2.53GHz x 4, GPU integrada Intel HD Graphics 3400, 5.6Gb de memória principal e 500Gb de memória secundária (HD). Tivemos dificuldade no arranjo da máquina para realização dos experimentos e testes portanto não foi realizado configurações avançadas de trial boot dos sistemas, por questões de praticidade para o grupo, cada sistema operacional teve exauridas quantidades de experimentos e testes por etapa na máquina.

## 1.2-Sistemas Operacionais Usados

**-Linux Ubuntu versão 20.04 LTS 64 bits;**

**-Solaris 11.4 com sistema de arquivos UFS 64 bits;**

**-FreeBSD 13 -RELEASE com sistema de arquivos UFS2 (Unix File System) 64bits.**

Retornando aos anos 1970, o UNIX original não era multithread, portanto não havia implementações compatíveis sobre os diversos SOs derivados do Unix. A família de normas POSIX foi criada para permitir essa compatibilidade. Então com a criação da biblioteca padrão unistr.h para a linguagem C, permitiu-se a adaptação de aplicações de linguagem C para máquinas Unix, possibilitando processamento multithread e a portabilidade para as diversas aplicações com padrões POSIX.

Assim como a biblioteca, os Sistemas Operacionais UNIX-like precisaram desenvolver seus próprios modelos de processamento de threads. Solaris e FreeBSD hoje utilizam o moderno modelo 1:1 (que utiliza apenas 1 thread kernel-level para 1 thread user-level, dando maior independência a todas as threads), já usaram o antigo modelo M:N (um modelo híbrido que vincula M threads user-level para N threads kernel-level, dependendo da situação), porém sua complexidade e seu custo computacional afetavam o desempenho do sistema. Todos os três sistemas têm suporte para a biblioteca unistd.h, mesmo que alguns tenham suas implementações próprias de threads (como as bibliotecas da linguagem C de threads, semáforos, etc. do próprio Solaris e seu compilador C do Solaris Studio), o que permitiu que o programa usado não precise ser modificado exclusivamente.



[illegible]

2- Tempo: 2.312 segundos

[illegible]

3- Tempo: 3.468 segundos

[illegible]

4- Tempo: 3.021 segundos

[illegible]

5- Tempo: 2.746 segundos

[illegible]

6- Tempo: 3.827 segundos

[illegible]

7- Tempo: 3.354 segundos

[illegible]

8- Tempo: 2.442 segundos

[illegible]

9- Tempo: 3.440 segundos

[illegible]

10- Tempo: 2.800 segundos

A média de tempo de execução é de **2.9935** segundos ( $\sum t/10$ , sendo  $\sum t$  o somatório dos tempos de execução)

## 2.2- Testes no Sistema Solaris

Executando os testes em um Sistema Solaris, tivemos os seguintes resultados:

[illegible]

1- Tempo: 3.014 segundos.

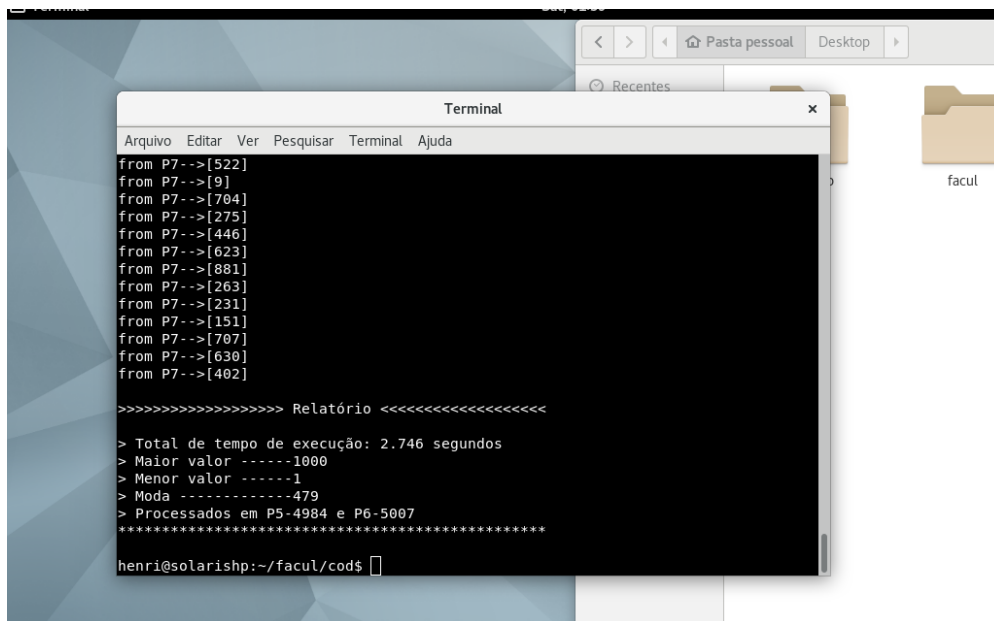










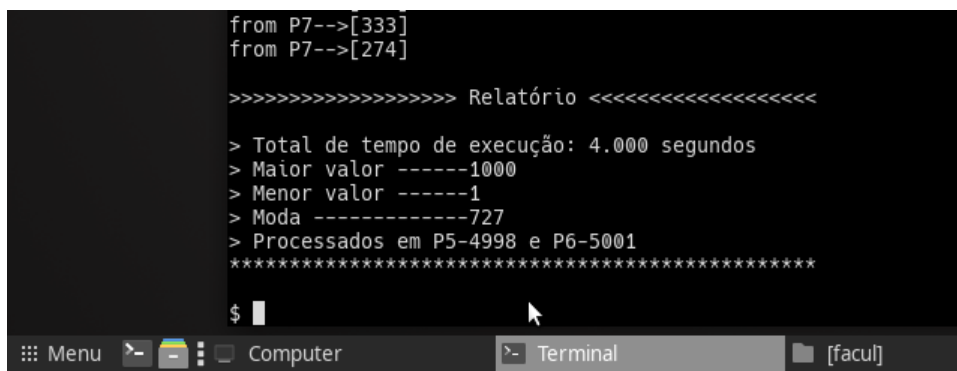


10- Tempo: 2.746 segundos.

A média de tempo de execução é de **2.5762** segundos ( $\Sigma t/10$ , sendo  $\Sigma t$  o somatório dos tempos de execução).

## 2.3- Testes no Sistema FreeBSD

Executando os testes em um Sistema FreeBSD, tivemos os seguintes resultados:

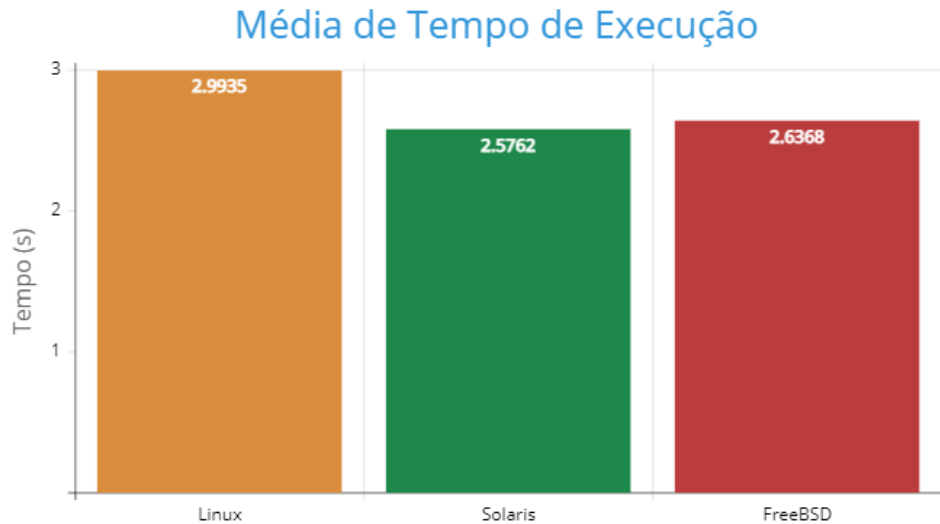


1- Tempo: 4.000 segundos.









### -Observações Adicionais

Dentre os três, Solaris e FreeBSD ambos são os mais antigos e mais confiáveis (apesar de FreeBSD ser mais novo que Linux, é uma “nova versão” de seu antecessor, conhecido como BSD ou Unix BSD, criado nos anos 1970). Solaris tem sistema de análise e resolução de problemas de performance em tempo real (Dtrace) e sua capacidade de prever e se antecipar a falhas e erros de execução antes que possam causar paradas de execução críticas (o chamado “Predictive Self-Healing”), que reduzem sua manutenção e o tornam essencial em Databases, Servidores e Cloud Computing.

O FreeBSD, tem sua estabilidade equiparável ao Solaris, muito usado para servidores DNS, provedores de internet e sites, não só por sua baixa manutenção e alta confiabilidade, licença de uso barata e seu Sistema de Ports, tornando-o tão ou até mais seguro que a maioria das distribuições Linux e preferível pelo baixo custo.

Já o GNU/Linux é considerado o kernel tipo Unix mais popular, compatível e versátil para diversos hardwares. Diferente dos anteriores, Linux tem diversas distribuições feitas por usuários sobre seu Kernel, de distribuições de uso geral, como Ubuntu, Fedora, para segurança da informação, como Kali ou até distribuições mobile, como o Android ou Ubuntu Touch. Seu uso abrange diversas áreas e por ser código aberto permite que terceiros criem suas distribuições particulares, como é o caso da Oracle Linux.



### 3-Conclusão

Com base nos testes executados podemos notar que o sistema Linux possuiu uma performance em média de 13,53% mais lento em relação ao FreeBSD e 16,20% mais lento que o Solaris na execução do ambiente multiprocessado. Como todos eles são Kernels monolíticos de mesma base Unix e que são preemptivos, podemos dizer que os SOs seguem caminhos bastante semelhantes para implementar os mesmos conceitos de processamento. O desempenho e escalabilidade dos três sistemas estão em níveis parelhos, bem implementados para suas principais áreas de aplicação. Porém, dos 3 sistemas operacionais, o Linux é o com maior tempo de execução neste caso. Para entender melhor, vamos olhar para o contexto de cada SO, o Linux possui um constante desenvolvimento de compatibilidade e frequentemente atualizado e adaptado para suportar várias funcionalidades e talvez seja por isso que ele é o mais moderno e utilizado de todos, mas isso tem em vista o objetivo de cada um, no quesito kernel e suas funcionalidades, apesar de serem bem compatíveis entre si, o Solaris por exemplo possui uma alocação de memória de alto desempenho, geralmente, por ser voltado para uso empresarial e Bancos de Dados, sua compatibilidade é um pouco maior para essas funcionalidades, apesar de ter uma maior restrição no quesito segurança entre troca de threads e processos. Além disso, em respeito à paginação o que o Linux chama de swapping, o Solaris pode fazer swap de processos inteiros, além de que suas estatísticas de processos são binárias, mais eficiente por padrão, e no caso do FreeBSD, ele compartilha abordagens semelhantes com o Solaris em seu design interno de processo de baixo nível, estruturas de thread e primitivas, distinto do Linux que tem seu próprio design de multithreading.