

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO

Laboratório de sistemas operacionais

Relatórios das práticas

André Marcelino de Souza Neves
andreneves3@gmail.com

Professor: Leonardo Lacerda Alves

Link do arquivo no overleaf: <https://www.overleaf.com/read/xvrkgphksygy>

Timóteo
2018

Sumário

1	GERENCIADOR DE PROCESSOS	5
1.1	Testes em Linux	5
1.1.1	Teste com editor de texto <i>Gedit</i>	5
1.2	Testes em Windows	6
1.2.1	Teste com editor de texto <i>Bloco de Notas (notepad)</i>	6
1.3	Sistemas Unix	7
1.4	Técnicas de escalonamento	7
1.4.1	Linux	7
1.4.2	Windows	7
1.4.3	Oracle Solaris (Unix)	7
2	MEMÓRIA VIRTUAL	8
2.1	Introdução	8
2.2	Desenvolvimento	8
2.3	Conclusão	9
3	SISTEMAS DE ARQUIVOS	10
3.1	Introdução	10
3.2	Desenvolvimento	10
3.3	Conclusão	11
4	IMPLANTAÇÃO	13
4.1	Introdução	13
4.2	Desenvolvimento	13
4.2.1	Implantação em máquina local	13
4.2.2	Custo financeiro em máquina local	14
4.2.3	Custo financeiro na <i>Google Cloud</i>	15
4.3	Conclusão	16
	REFERÊNCIAS	18

Introdução

Este documento contém os relatórios realizados nas aulas práticas de **Laboratórios de Sistemas Operacionais**, que teve como objetivos avaliar informações e comportamentos dos módulos de **Gerenciadores de Processos**, **Memória Virtual** e **Sistemas de Arquivos**. Além disso, foram realizadas duas práticas que tinham como foco os aspectos de sistemas operacionais para implantação de aplicações para *web*, com análise de custo para manutenção de um sistema local/caseiro e em nuvem pública.

A análise do **gerenciadores de processos** foi realizada na distribuição *Manjaro* na versão 18.0.4, e *kernel* na versão 4.19.62-1-MANJARO, além de ser realizada também em *Windows*, por meio de uma máquina virtual que contém uma distribuição simplificada, oferecida pela *Microsoft* em seu site ¹. A distribuição escolhida foi a do *Windows 7 Enterprise* na versão 6.1.7601 Service Pack 1 Build 7601, com *Internet Explorer 11*.

Nesta prática foi possível observar alguns aspectos de gerenciamento de processos em **Linux** e **Windows**, como: Método de visualização da lista de processos ativos, identificadores de processos, características do primeiro processo ativo no sistema, usuários donos de processos, gerenciamento da execução de processos (pausa, continuação e finalização) e técnicas de escalonamento em cada um dos sistemas.

As práticas de **memória virtual** e **sistemas de arquivos** foram realizadas com a distribuição **Mint** do linux na versão 19.2 Tina. A prática de **Memória virtual** teve como objetivo manipular o recurso no *Linux*, de forma a verificar as configurações atuais e acrescentar capacidade de memória, além de discutir sobre a eficiência do recurso quando utilizado em cima de **Arquivos** ou de **Partições Swap**, o que é possível em sistemas *Linux*. Além disso, também foi discutido sobre como o recurso funciona em sistemas *Windows*.

A prática de sistemas de arquivos teve como objetivo simular um processo de auditoria de um dispositivo de armazenamento, que consiste em gerar uma imagem de um fragmento do disco e analisar o conteúdo por meio de um editor de textos hexadecimal.

As práticas de implantação de aplicações web foi realizada com o objetivo de instalar uma loja funcional com a plataforma Prestashop (2019a). A prática teve duas etapas: A primeira foi análise e instalação do *prestashop* em um computador local/caseiro, e a segunda foi análise de custo de manutenção da mesma aplicação em uma nuvem pública dos provedores mais conhecidos (Google, Amazon ou Microsoft).

O sistema local/caseiro foi testado em *linux* com distribuição **Ubuntu Server** na versão 18.03.1.3 LTS Release 18.04 com *kernel* na versão 4.15.0-66-generic. O sistema foi executado em uma máquina virtual com o software *Virtual Box* na versão 6.0. A máquina hospedeira tinha um processador AMD A10 pro-7800b r7, com 6,8 GB de memória RAM, sendo que dessa capacidade, foram reservados 512 MiB de memória RAM para a máquina virtual.

¹ <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>

Todas as práticas tiveram como foco os aspectos relacionados ao sistema operacional *Linux*, mas não necessariamente na mesma distribuição ou ambiente. Também foram realizadas algumas avaliações e comparações para o sistema *Windows*.

1 Gerenciador de processos

1.1 Testes em Linux

Para esta prática foi utilizado o Linux com distribuição *Manjaro* na versão 18.0.4, e com o *kernel* na versão 4.19.62-1-MANJARO. As informações foram obtidas por meio da execução dos comandos `lsb_release -a` e `uname -r`, que fornecem, respectivamente, a versão da distribuição e do *kernel* (OLIVEIRA, 2018).

O comando `ps` permite obter informações sobre processos no sistema, e exibe informações como usuário que executou, identificador do processo e horário de início de execução. As variações `ps aux` e `ps -A` exibem os processos de todos os usuários do sistema, em vez de mostrar apenas processos do usuário / seção atual (Computer Networking Notes, 2019).

Com a execução de ambos os comandos, foram obtidas as seguintes informações no início da lista, conforme blocos 1.1 e 1.2:

Listing 1.1 – Comando `ps aux` (resumido)

USER	PID	COMMAND
root	1	/sbin/init

Listing 1.2 – Comando `ps aux` (resumido)

PID	CMD
1	systemd

É possível perceber que estas linhas referem-se ao primeiro processo inicializado no sistema, pois possuem *PID - Process ID - Identificador de Processo* igual a 1, e são identificados como `/sbin/init` ou `systemd`. Por estar listado no resultado desse comando, significa que o processo continua em execução.

Com o comando `ps aux` é possível identificar os donos dos processos, que são usuários que os inicializaram. Os nomes dos usuários são listados na primeira coluna da lista (**USER**). Os usuários listados foram: **root**, **andre** (Meu usuário), **gdm** (Gnome Display Manager (ArchWiki, 2019)), **colord**, **rtkit**, **postgres**, **dbus** e **systemd**+

1.1.1 Teste com editor de texto *Gedit*

A segunda baterias de testes no linux foi realizada com o editor de textos *Gedit*. A inicialização é feita por meio do comando de inicialização do programa, `gedit`. Para determinar a identificação do processo do editor de texto, foi utilizado o comando `pidof gedit`, que permite conhecer todas as instâncias de processo de determinado programa. A identificação também poderia ser feita com o comando `ps aux`, sendo essa uma forma mais trabalhosa por exigir busca na lista de vários outros processos. O número do processo do editor de texto foi 27878

É possível alterar a prioridade de execução de um processo no linux por meio do comando *renice [nice value] -p [PID]*, que deve ser executado com permissões de administrador. O parâmetro *nice value* indica a ordem de prioridade, um valor entre -20 e 19, onde o menor número atribuir maior prioridade (NixTutor, 2009). Com isso, foi dada prioridade máxima com utilização do comando: *sudo renice -20 -p 27878*

O comando *kill* é utilizado para manipular o ciclo de vida de um processo com as operações de encerramento, interrupção temporária e continuação de execução. O comando *kill -STOP [PID]* interrompe temporariamente o processo, deixando-o travado, enquanto *kill -CONT [PID]* retoma a execução. Para encerrar na força bruta é utilizado o comando *kill [PID]* (OS Tech Nix, 2017).

1.2 Testes em Windows

Os testes em *Windows* foram realizados em uma máquina virtual que contém uma distribuição simplificada oferecida pela *Microsoft* em seu site ¹. A distribuição escolhida foi a do *Windows 7* com *Internet Explorer 11*.

Com o comando *systeminfo* (SANTANA, 2012), foi possível obter as informações detalhadas do sistema operacional. A distribuição usada é *Microsoft Windows 7 Enterprise*, na versão *6.1.7601 Service Pack 1 Build 7601*.

O comando *tasklist* (Microsoft, 2017) permite obter informações sobre os processos em atualmente no computador local ou em um computador remoto. O comando exibe uma tabela que informa o nome do executável do processo, sua identificação, nome e número da seção e uso de memória.

Por ter número de identificação igual à 0, automação infere-se que o processo de nome *System Idle Process* foi o primeiro a ser executado no sistema. Esse comando não exibe o nome do usuário dono do processo, e para identificá-lo foi necessário abrir o **Gerenciador de Tarefas** na aba **Processos**

1.2.1 Teste com editor de texto *Bloco de Notas (notepad)*

Esse momento de testes do *windows* foi realizado com o editor de texto *Bloco de notas*. A identificação da identificação do processo foi feita com utilização do comando *tasklist*. Para facilitar o trabalho, é possível usar os filtros do comando, de forma a exibir apenas os processos inicializados pelo bloco de notas, com o seguinte comando: *tasklist /fi "IMAGENAME eq notepad.exe"*.

O controle de prioridade é feito próprio **Gerenciador de Tarefas**, na aba **Processos**. Ao clicar com o botão direito, existe a opção de alterar a prioridade (Tempo real, alta, acima do normal, normal, abaixo do normal, e baixa).

Com o **Gerenciador de Tarefas** também é possível encerrar na força bruta um processo. Na aba **Processos**, basta selecionar um item e clicar em "Finalizar Processo". Contudo, para as opções de pausar e retomar a execução, torna-se necessário o uso da ferramenta *Resource Monitor*, acessada

¹ <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>

pelo comando *resmon*. Na seção *CPU*, ao clicar com o botão direito, existem as opções de pausar e continuar a execução do processo (WinDroidWiz, 2019).

1.3 Sistemas Unix

As operações realizadas nos testes acima podem ser feitas em sistemas *Unix* de forma semelhante aos procedimentos em *Linux*, pois os comandos utilizados possuem mesma sintaxe em ambos os casos. Os comandos utilizados foram: **ps** (Shapes Hed, 2016), **pidof** (Tutorials Point, 2019), **renice** (Computer Hope, 2019) e **kill** (OS Tech Nix, 2017).

1.4 Técnicas de escalonamento

1.4.1 Linux

O *Linux* utiliza como algoritmo de escalonamento o *Completely Fair Scheduler (CFS)*, que baseia-se no princípio de provisionar de forma balanceada tempo de processador para as tarefas, ou seja, todas as tarefas devem, dentro de um mesmo intervalo de tempo, obter exatamente o mesmo tempo de processador (Project CSCI, 2019).

Quando tal situação não ocorre, o algoritmo considera que há uma falta de equilíbrio nas execuções, as tarefas que receberam menos tempo de processador são tomadas com maior prioridade, de forma a manter a execução equilibrada. Para organização do tempo de execução das tarefas, o algoritmo utiliza uma árvore *Red-Black (Rubro-Negra)*.

1.4.2 Windows

Nos sistemas Windows-NT, é utilizado o algoritmo *MLFQ (Multilevel Feedback Queue)*, que defini 32 níveis de prioridade, divididos em grupos. O usuário pode selecionar apenas o grupo de prioridade desejado, e o sistema faz o ajuste fino conforme o uso de I/O, CPU e se o processo é interativo (responde à comandos de interface homem-máquina). Nesse último caso, o processo ganha maior prioridade, para garantir maior responsividade (Wikipedia, 2019). Nos escalonadores do tipo *Multilevel*, cada nível de prioridade possui sua própria fila com escalonamento *round-robin* nas filas de maiores prioridades *FIFO* nas filas de menor prioridade.

1.4.3 Oracle Solaris (Unix)

O sistema *Oracle Solaris* foi escolhido dentre os sistemas *Unix*, para estudo da técnica de escalonamento de processos. O escalonador padrão é o *timesharing scheduler*, o qual provisiona equidade de acesso aos recursos de *CPU* (Oracle, 2012b), com alocação de tempo de acordo com a prioridade. Também é possível utilizar o *Fair Share Scheduler*, o qual garante uma justa alocação de recursos de *CPU* entre os processos (Oracle, 2012a). A garantia de justiça é feita de modo semelhante ao algoritmo *Completely Fair Scheduler*, atualmente usado em *linux*.

2 Memória virtual

2.1 Introdução

Esta prática tem como objetivo analisar um importante recurso em sistemas operacionais, que é a **memória virtual**. Esse artifício permite utilização do disco rígido como paliativo à escassez de memória RAM (Elaine Martins, 2012). Para esta prática foi utilizado o Linux com distribuição *Mint* na versão 19.2 Tina, e com o *kernel* na versão 4.15.0-54-generic.

2.2 Desenvolvimento

Inicialmente, foram verificadas as informações sobre uso de memória virtual pelo sistema, por meio do comando **swapon -s** (Nix Craft, 2019). De acordo com o retorno, o sistema utiliza apenas memória virtual armazenada no arquivo `/swapfile`, de tamanho *989,81 MB*.

A criação do arquivo de memória virtual é feita com o utilitário *dd* (KILI, 2017), com o comando do bloco 2.1:

Listing 2.1 – Comando *dd* para criar o arquivo de memória virtual. O parâmetro **bs** representa o tamanho do bloco de leitura simultânea, e **count** é a multiplicação de **bs** pelo número de *Megabytes* do arquivo, nesse caso, 100 MB.

```
sudo dd if=/dev/zero of=/mnt/swapadicional bs=1024 count=102400
```

Após isso, os seguintes passos foram realizados:

- Tornar o arquivo acessível apenas para *root*: **sudo chmod 600 /mnt/swapadicional**.
- Definí-lo como *swap*: **sudo mkswap /mnt/swapadicional**.
- Habilitar seu uso como arquivo de *swap*: **sudo swapon /mnt/swapadicional**

Após seguir os passos acima, foi executado novamente o comando **swapon -s**, e verificado que o novo arquivo está disponível na listagem:

Listing 2.2 – Comando **swapon -s**, após criado o novo arquivo de memória virtual.

Filename	Type	Size (KB)	Used(KB)	Priority
/swapfile	file	1013564	35840	-2
/mnt/swapadicional	file	102396	0	-3

Após reiniciar o sistema, foi verificado pelo comando **swapon -s** que o arquivo criado para memória virtual não existe mais. Isso aconteceu porque não foi feita nenhuma determinação para que o arquivo fosse montado como *swap* no momento do *boot*.

2.3 Conclusão

Nessa prática foi realizada a criação e utilização de um novo arquivo para ser usado como **Swap**. Em sistemas **Linux**, é possível configurar memória virtual com utilização de arquivos ou de partições com o sistema de arquivos **Swap**. A utilização de memória virtual em uma partição tende a ser a opção com mais desempenho, pois nesse caso é utilizado um sistema de arquivos totalmente preparado e específico para uso como memória virtual, o que evita o *overhead* existente nos sistemas de arquivos convencionais, como **ext4** ou **NTFS** (Quora, 2014). Esse *overhead* existe devido ao contexto em que os sistemas convencionais estão inseridos, onde é necessário ter gerência de vários arquivos, com diferentes permissões de usuários e controle de índices.

No **Windows**, não existe opção para utilizar memória virtual como partição, e todo o recurso é concentrado no arquivo *pagefile.sys* (BRITO, 2016), e com isso, toda as questões de formato e desempenho discutidas no parágrafo anterior são sempre existentes para sistemas **Windows**.

3 Sistemas de arquivos

3.1 Introdução

Esta prática tem como objetivo avaliar um dos aspectos que envolvem os **sistemas de arquivos** de um sistema operacional, que é a auditoria externa de dispositivos de armazenamento. Nessa situação, foi necessário copiar e analisar parte do sistema de arquivos de um sistema **Windows**. Para esta prática foi utilizado o Linux na distribuição *Mint* na versão 19.2 Tina, e com o *kernel* na versão 4.15.0-54-generic, além de uma instalação simplificada do *Windows 7* com *Internet Explorer 11*, disponibilizada pela *Microsoft* em seu site ¹.

3.2 Desenvolvimento

Inicialmente, foi criada uma nova máquina virtual, na qual foi anexado um novo disco rígido auxiliar, um disco de *Live Boot* do linux utilizado e o disco que continha a instalação do *Windows 7*. O disco auxiliar foi formatado para o sistema de arquivos NTFS com ajuda do utilitário **Gerenciador de Discos** do *Windows* (TecMundo, 2013).

Para verificar os discos e partições identificados pelo sistema, foi utilizado o comando **sudo fdisk -l** (Blog do Cardoso, 2015), com o qual é possível identificar o caminho usado pelo sistema operacional. Uma parte da saída do comando, nessa situação, é mostrada no bloco 3.1. Foi possível identificar que o primeiro disco tem caminho `/dev/sda`, e que o disco auxiliar possui o caminho `/dev/sdb`.

Listing 3.1 – Demonstração de saída do comando **sudo fdisk -l**.

```
Disk /dev/sda: 28.8 GiB, 30945099776 bytes, 60439648 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xecc58615
```

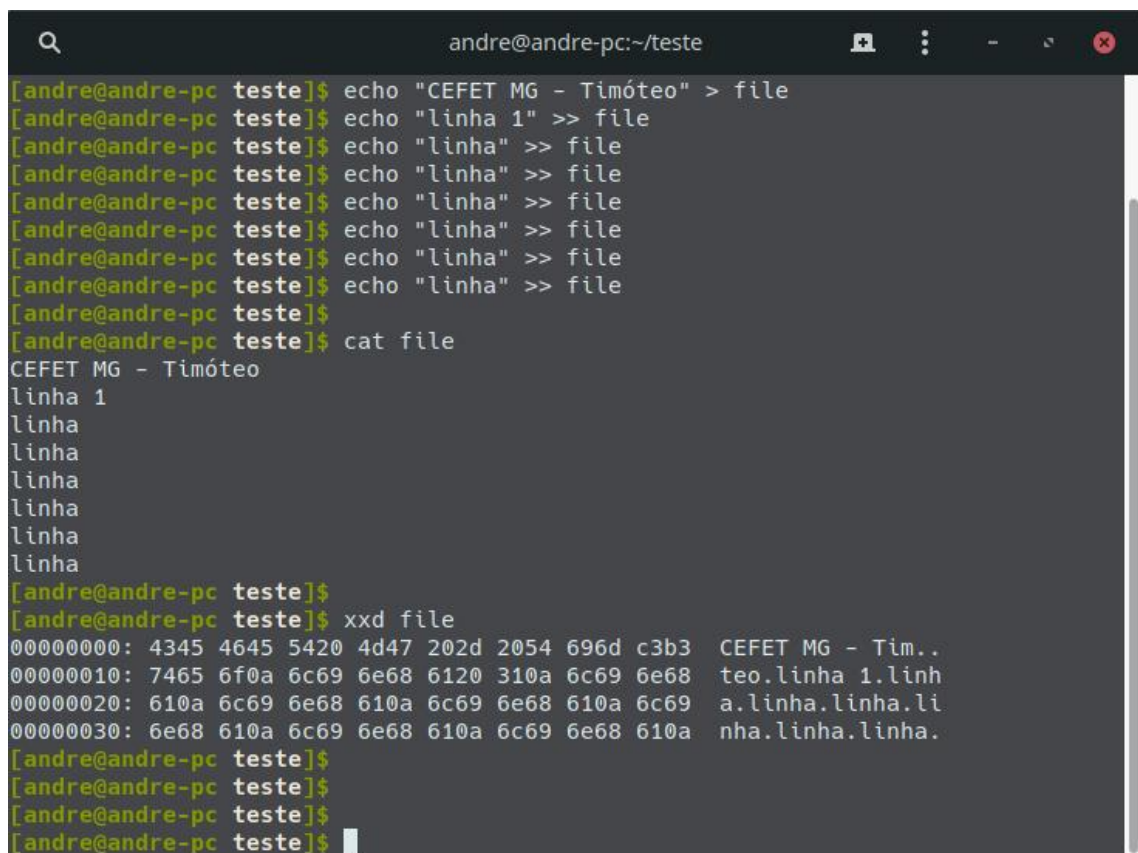
Com o utilitário **dd**, foi copiado um segmento da partição que continha a instalação do *Windows* de testes (Matt Vas, 2019). O processo foi executado por 15 minutos e realizou a cópia de um volume de 14,35 GB. O comando utilizado foi: **sudo dd if=/dev/sda of=/dev/sdb**.

Na tentativa de montar e gerar a imagem da partição para a qual a cópia foi feita, foi executado o seguinte comando: **sudo mount /dev/sdb1**. Um erro ocorreu nesse momento: **mount: /mnt: wrong fs type, bad option, bad superblock on /dev/sdb1**. Nesse contexto, não foi possível gerar o arquivo de imagem da cópia dentro da nova partição.

¹ <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>

No entanto, ao inserir o disco virtual do *Windows* em uma máquina virtual com o sistema *linux* já instalado, foi possível gerar a imagem do disco original diretamente dentro do disco de sistema, no diretório `/home`. Foi feita a seguinte execução: `sudo dd if=/dev/sda of=/home/arquivo.img`.

A análise do conteúdo do disco original foi feita com o utilitário `xxd`, o qual exibe o conteúdo de um arquivo em formato hexadecimal (Shahriar Shovon, 2018). Não foi possível encontrar qualquer referência a respeito de algum possível diretório existente no sistema de arquivos de origem, no entanto um teste à parte foi realizado no momento. Na figura 1, é mostrada uma sequência onde um arquivo com conteúdo aleatório foi criado, e teve seu conteúdo auditado com utilização do editor de textos hexadecimal. Verificou-se que o conteúdo é perfeitamente auditável.

A terminal window titled 'andre@andre-pc:~/teste' shows a series of commands. First, a file named 'file' is created using 'echo'. Then, the file's content is displayed using 'cat', showing a list of lines. Finally, the file's content is audited using 'xxd', which displays the hexadecimal representation of the text. The audit output shows the file's content in hexadecimal, with some lines truncated for brevity.

```
andre@andre-pc teste]$ echo "CEFET MG - Timóteo" > file
andre@andre-pc teste]$ echo "linha 1" >> file
andre@andre-pc teste]$ echo "linha" >> file
andre@andre-pc teste]$ echo "linha" >> file
andre@andre-pc teste]$ echo "linha" >> file
andre@andre-pc teste]$ echo "linha" >> file
andre@andre-pc teste]$ echo "linha" >> file
andre@andre-pc teste]$ cat file
CEFET MG - Timóteo
linha 1
linha
linha
linha
linha
linha
linha
andre@andre-pc teste]$
andre@andre-pc teste]$ xxd file
00000000: 4345 4645 5420 4d47 202d 2054 696d c3b3  CEFET MG - Tim..
00000010: 7465 6f0a 6c69 6e68 6120 310a 6c69 6e68  teo.linha 1.linh
00000020: 610a 6c69 6e68 610a 6c69 6e68 610a 6c69  a.linha.linha.li
00000030: 6e68 610a 6c69 6e68 610a 6c69 6e68 610a  nha.linha.linha.
andre@andre-pc teste]$
andre@andre-pc teste]$
andre@andre-pc teste]$
andre@andre-pc teste]$
```

Figura 1 – Demonstração do utilitário `xxd` em um arquivo de texto.

Também foi feito um teste de auditoria do conteúdo do disco `/dev/sda`, conforme mostra a figura 2.

3.3 Conclusão

Com essa prática, foi possível simular um processo importante em várias situações, onde é necessário realizar auditoria de um dispositivo de armazenamento.

Esse processo não deve ser feito com a utilização do sistema operacional instalado no próprio dispositivo auditado, pois podem existir armadilhas que impedirão o processo. Ao invés disso, o dispositivo pode ser conectado a um computador com um sistema operacional **linux** e ter todos os dados existentes clonados para uma imagem do disco por meio do utilitário `dd`.

```

andre@andre-pc:~
000029f0: 28aa f2fc d8e8 52e4 1c9c 6dcb 2a26 2a98 (.....R...m.*&*.
00002a00: 4e43 a387 e6a2 d78e 25ee 5c11 4ba5 6000 NC.....%.\.K.`.
00002a10: 904c 1c50 10a2 f3bc d8cb d974 ee85 1601 .L.P.....t....
00002a20: d2db a906 dabc 4046 c401 c27e 7a3d bdb1 .....@F....~z=..
00002a30: 7329 0d23 b96e c79e f2f2 af15 746b fb2b s).#.n.....tk.+
00002a40: 398e 268e dc33 85d0 ec67 237b 0046 a839 9.&...3...g#{.F.9
00002a50: fee9 6be9 5284 529f bcba 3b22 d95f ab56 ..k.R.R...;"._.V
00002a60: 02e9 be63 9538 558c ad12 d791 da5f 2f4e ...c.8U....._/N
00002a70: f854 5528 478f d690 5a75 056a 6a1f 1e69 .TU(G...Zu.jj..i
00002a80: 0f7f 4fd0 165c f815 464b 4dae 13ed c02e ..O..\..FKM.....
00002a90: d36e de8c 7e43 0ac1 ecaa d5c5 2dea 9a6d .n...~C.....-..m
00002aa0: 5002 6110 5748 c646 d1f6 17b9 685e f25c P.a.WH.F....h^.\
00002ab0: 6f8f b5c9 12a3 b467 a8e9 2979 3605 9e49 o.....g..)y6..I
00002ac0: fc82 020b 67b2 a168 5e7d 24c0 2a63 2ffc ...g..h^}$.*/.
00002ad0: 3280 5f8f a950 2184 dd8f 1edd a71c aaf6 2._..P!.....
00002ae0: ce45 6c33 eaf3 7d5d 18fb df8a c202 0ac9 .El3..}].....
00002af0: 0228 4ddc c111 f370 4217 e57c 2c2a b1cc .(M....pB...|,*..
00002b00: 5f2c b2a4 9a9c 5f51 ad82 f03d 6aea b879 _,..._Q...=j..y
00002b10: 4940 315b e450 e845 5d99 dab0 02bc 87d5 I@1[.P.E].....
00002b20: 7fbf 9ab1 816d 59b0 d6e2 5bef 028e c388 .....mY...[.....
00002b30: fb60 8f68 c44d fad8 66b4 f22a 5bc6 025f .`.h.M..f..*[_..
00002b40: dd52 6dfb 1ee0 60cd d318 5cf8 8fa4 c06a .Rm...`....\....j
^C
[andre@andre-pc ~]$
[andre@andre-pc teste]$
[andre@andre-pc teste]$
[andre@andre-pc teste]$
[andre@andre-pc teste]$

```

Figura 2 – Demonstração do utilitário `xdd` no disco `/dev/sda`.

Com o utilitário `xdd` é possível realizar uma busca por informações que estavam armazenadas no dispositivo original. É importante ressaltar que esse processo não seria tão simples caso o sistema de arquivos fosse criptografado. Nessa situação é necessário recorrer a um método de descriptografia do sistemas de arquivos, o que torna o processo de auditoria muito mais trabalhoso.

4 Implantação

4.1 Introdução

Desde a criação da internet, o acesso à serviços e aplicações *online* passou a fazer parte do uso dos computadores, e cada vez mais serviços são criados para proporcionar aos usuários novas possibilidades de exploração no mundo digital. Alguns exemplos desses serviços são: Redes sociais, lojas virtuais, sistemas acadêmicos e portais de notícias.

No entanto, manter tais serviços em correto funcionamento depende de uma infraestrutura computacional adequada, o que significa possuir recursos contra os problemas que podem acontecer, como falta de energia, problemas de conexão com internet, sistema de refrigeração inadequado e falta de segurança física (Infotrend Brasil, 2019). Garantir os requisitos básicos de segurança, confiabilidade e disponibilidade demandam alto custo e mão de obra especializada, o que dificulta a execução da operação de forma correta.

Para isso, é possível recorrer ao uso de computação em nuvem, com atenção inicial para os produtos de *IaaS* (*Infrastructure as a Service - Infraestrutura como serviço*). Segundo EVEO (2017), os *IaaS's* trazem o conceito de *Data Center* virtual, no qual as empresas podem montar toda sua infraestrutura computacional por meio de um painel de controle, de forma que não dependa de manutenção de *data centers* e máquinas físicas. As soluções de *Data Centers* Virtuais são vendidas sob medida, de acordo com a necessidade de cada cliente, e com isso permite trocar todo investimento que seria feito em um *data center* físico por uma terceirização de manutenção.

4.2 Desenvolvimento

Esta prática foi dividida em duas partes: A primeira está descrita nas seções 4.2.1 e 4.2.2 tinha como objetivo implantar e avaliar o custo financeiro de manutenção do serviço *PrestaShop* em um computador caseiro. Na segunda parte está descrita na seção 4.2.3, e apresenta a análise do custo financeiro para manutenção do serviço na infraestrutura do *Google Cloud*.

4.2.1 Implantação em máquina local

Inicialmente, foi preparada um servidor *web* em uma máquina virtual com uma máquina hospedeira com processador *AMD A10 pro-7800b r7* e 6,8 GB de memória RAM, sendo que dessa capacidade foram reservados 512 MB de memória RAM para a máquina virtual.

Na máquina virtual foi utilizado o Linux com distribuição **Ubuntu Server** na versão **18.031.3 LTS Release 18.04** com *kernel* na versão **4.15.0-66-generic**. As informações foram obtidas por meio da execução dos comandos *lsb_release -a* e *uname -r*, que fornecem, respectivamente, a versão da distribuição e do *kernel* (OLIVEIRA, 2018).

O consumo basal observado após a instalação do sistema foi obtido por meio do utilitário *htop* do linux, e está mostrado na tabela 1.

Memória	SWAP	CPU
119 MB	12,9 MB	0.0%

Tabela 1 – Consumo basal após a instalação do sistema

Após a observação do consumo basal do sistema operacional, foi seguido o tutorial do site DigitalOcean (2016) para instalação das aplicações necessárias para funcionamento do *prestashop*, que são (Prestashop, 2019b):

- *Apache Web Server* na versão 2.2 ou superior;
- *PHP* na versão 7.1 ou superior;
- *MySQL* na versão 5.6 ou superior

Com isso, o consumo basal foi observado novamente, com os dados apresentados na tabela 2.

	Memória	SWAP	CPU
Valor	316 MB	58,9 MB	1.3%
Aumento	197 MB	46 MB	1,3%

Tabela 2 – Consumo basal após a instalação do Apache + PHP + MySQL

Após isso, foi feito *download* e instalação do *prestashop* no servidor local, e o consumo basal foi novamente medido, agora com as informações apresentadas na tabela 3

	Memória	SWAP	CPU
Valor	366 MB	146,5 MB	10%
Aumento	50 MB	87,6 MB	8,7%

Tabela 3 – Consumo basal após a instalação do *Pretashop*

O aumento do consumo basal deve-se à necessidade de ativação dos serviços *Apache*, *PHP* e *MySQL*. Com a instalação do *prestashop*, muitas tabelas foram criadas no banco de dados, de forma que é necessário um maior custo computacional para gerenciamento das rotinas de manutenção do banco.

4.2.2 Custo financeiro em máquina local

Após realizados os testes, foi estimado o custo financeiro mensal de manter o serviço ativo de forma ininterrupta. Para isso, foi considerado os três seguintes objetos:

- Computador físico: Para manter o serviço, foi avaliado um computador minimamente compatível com a máquina virtual utilizada, em termos de memória RAM. Com isso, foi considerado o computador **Intel Dual Core 2.58gb 2gb 500gb Easypc Smart** ¹, que no momento de elaboração desse relatório tinha o custo de R\$ 817,52.

¹ <<https://www.americanas.com.br/produto/65206349/computador-com-monitor-led-intel-dual-core-2-58gb-2gb-500gb-easypc-sr>>

- Internet 100 Mbps para *download* e 50 Mbps para *upload*, fornecida pela giganet ²: R\$ 99,90.
- Energia elétrica convencional fornecidas pela CEMIG, na sua cotação mais cara ³: R\$ 0,62833 por kWh.

Para o cálculo, são feitas as seguintes considerações:

- Computador físico: Estima-se que um computador tenha uma duração, em boas condições, por 5 anos. Com isso, o valor mensal é dado pela respectiva fração de um mês desse período de tempo, dado pela equação $C = V \cdot \frac{1}{5} \cdot \frac{1}{12}$, onde V é o preço de compra, e C é o custo mensal.
- Energia elétrica: É necessário saber o consumo médio em Watts (W) de um computador. Para isso, foram consideradas as informações de Oliveira (2019), das quais pode-se concluir que o consumo médio é de $75,5 W$. Um funcionamento mensal ininterrupto representa um total de $30 \cdot 24 = 720$ horas. Com a consideração do consumo médio informado acima, o consumo elétrico mensal será de $75,5 \cdot 720 = 54360Wh = 54,360 kWh$

Com isso, os valores do cálculo de custo financeiro mensal estão apresentados na tabela 4

Item	Valor
Computador	$C = V \cdot \frac{1}{5} \cdot \frac{1}{12} = 817,52 \cdot \frac{1}{5} \cdot \frac{1}{12} = R\$ 13,63$
Internet	R\$ 99,90
Energia elétrica	$R\$ 0,62833 \cdot 54,360kWh = R\$ 34,16$
Total	R\$ 147,69

Tabela 4 – Custo de manutenção do *Prestashop* em máquina caseira

4.2.3 Custo financeiro na *Google Cloud*

Para essa prática, foi utilizada a plataforma do *Google Cloud*, que fornece soluções em nuvem para diversas finalidades. A plataforma de nuvem fornece um *marketplace* no qual é possível ter acesso à diversos pacotes de serviços já configurados, e prontos para serem implantados e utilizado. Nesse caso, foi utilizado o pacote de instalação do *PrestaShop* na *Google Cloud* certificado pela *Bitnami* ⁴. O pacote possui as seguintes características:

- Sistema computacional:
 - Sistema Operacional *Debian 9*
 - Memória: 1,7 GB
 - Processador: 1 vCPU compartilhada (g1-small)
- *Softwares* instalados:

² <<https://giganetdc.com.br/>>

³ <https://www.cemig.com.br/pt-br/atendimento/Paginas/valores_de_tarifa_e_servicos.aspx>

⁴ <<https://console.cloud.google.com/marketplace/details/bitnami-launchpad/prestashop>>

- Apache 2.4.41
- lego 3.1.0
- MySQL 8.0.17
- OpenSSL 1.1.1d
- PHP 7.2.23
- phpMyAdmin 4.9.1
- PrestaShop 1.7.6.1
- SQLite 3.30.0.
- Varnish 6.0.4

De acordo com as características de *hardware* especificadas pelo pacote, a implantação possui custo mensal de \$ 13,61. Com a utilização da cotação do dólar no momento da finalização deste relatório (R\$ 4,17), a solução possui custo mensal de R\$ 56,75

4.3 Conclusão

Com a finalização dessa prática foi possível ter uma noção geral a respeito do custo de manutenção de um serviço fornecido pelo *PrestaShop*, seja em infraestrutura independente ou no *Google Cloud*.

A máquina física considerada no orçamento possui maior quantidade de memória RAM (2 GB) do que o consumo basal final, que fora observado após a instalação do serviço *PrestaShop*. No entanto, foi escolhida por ser uma quantidade mínima recomendada para a maioria dos sistemas computacionais, e além disso, serão necessários mais recursos do que os observados no consumo basal.

A solução fornecida pela *Google Cloud* possui capacidade de memória RAM aproximada em relação à máquina avaliada para compra, o que torna mais justa a comparação. Além disso, por ser uma máquina incluída em um pacote específico para o *PrestaShop*, acredita-se que seja um *setup* suficiente para uso inicial do serviço.

De acordo com os cálculos, nota-se que o valor financeiro da implantação em nuvem é muito mais vantajoso em relação à montagem de um *setup* caseiro, tendo em vista que a primeira solução apresenta uma economia de 61,57 % do valor monetário. Além disso, a solução em nuvem mostra-se muito mais vantajosa por outros aspectos que são garantidos, como segurança e disponibilidade, conforme citado por EVEO (2017) e discutido no início desse capítulo.

Conclusão final

As práticas realizadas para a disciplina de **Laboratórios de Sistemas Operacionais** permitiram um estudo e manipulação dos principais aspectos que envolvem o gerenciamento de sistemas computacionais de pequeno e grande porte. Por exemplo: O entendimento e correta configuração de **escalonadores de processos** resultam em diferentes consequências operacionais, principalmente em se tratando de servidores de grande porte, que necessitam responder à muitas requisições simultâneas. Além disso, os conceitos aprendidos e praticados nos tópicos de **Memória Virtual** e **Sistemas de arquivos** são de grande importância para garantir performance e estabilidade.

Em especial, os conhecimentos de **Sistemas de Arquivos** são essenciais para que os administradores de sistemas tenham maior controle da persistência de dados importantes para os usuários, desde o momento em que estão em armazenamento ativo, pronto para ser lido, modificado ou apagado, ou em situações onde os dados estão armazenados em discos de *backups* ou que foram apreendidos para inspeção e auditoria. Em todas essas situações, é necessário ter procedimentos adequados de manipulação das unidades, de forma a evitar a perda dos dados.

As três práticas iniciais trataram de assuntos relacionados à administração básica dos sistemas operacionais. Para além desses tópicos, foi realizado nas últimas duas seções a análise de implantação de uma aplicação *web*, feitas em um servidor caseiro ou em nuvem. Esta última teve como objetivo comparar as duas opções de implantação, com uma análise própria de custo de manutenção, e levando em conta aspectos relevantes como: Confiabilidade, disponibilidade e segurança.

Referências

ArchWiki. *GDM (Português) - ArchWiki*. 2019. Disponível em: <[https://wiki.archlinux.org/index.php/GDM_\(Portugu%C3%AAs\)](https://wiki.archlinux.org/index.php/GDM_(Portugu%C3%AAs))>. Citado na página 5.

Blog do Cardoso. *Linux: comandos para verificar partições de disco* Blog do Cardoso. 2015. Disponível em: <<https://blogdocardoso.com/linux-comandos-para-verificar-particoes-de-disco/>>. Citado na página 10.

BRITO, E. *O que é pagefile.sys e como usá-lo para melhorar o Windows 10 | Dicas e Tutoriais | TechTudo*. 2016. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2016/09/o-que-e-pagefilesys-e-como-usa-lo-para-melhorar-o-windows-10.html>>. Citado na página 9.

Computer Hope. *Linux renice command help and examples*. 2019. Disponível em: <<https://www.computerhope.com/unix/renice.htm>>. Citado na página 7.

Computer Networking Notes. *ps aux command and ps command explained*. 2019. Disponível em: <<https://www.computernetworkingnotes.com/linux-tutorials/ps-aux-command-and-ps-command-explained.html>>. Citado na página 5.

DigitalOcean. *Como instalar a pilha Linux, Apache, MySQL, PHP (LAMP) no Ubuntu 16.04 | DigitalOcean*. 2016. Disponível em: <<https://www.digitalocean.com/community/tutorials/como-instalar-a-pilha-linux-apache-mysql-php-lamp-no-ubuntu-16-04-pt>>. Citado na página 14.

Elaine Martins. *O que é memória virtual? - TecMundo*. 2012. Disponível em: <<https://www.tecmundo.com.br/o-que-e/1031-o-que-e-memoria-virtual-.htm>>. Citado na página 8.

EVEO. *Como o cloud computing pode ajudar na redução de custos com data center? - Blog da EVEO Cloud*. 2017. Disponível em: <<https://www.eveo.com.br/blog/cloud-computing-reducao-custos/>>. Citado nas páginas 13 e 16.

Infortrend Brasil. *Requisitos para construir seu próprio Data Center*. 2019. Disponível em: <<http://www.infortrendbrasil.com.br/dicas/montar-data-center-na-empresa/>>. Citado na página 13.

KILLI, A. *How To Create a Linux Swap File*. 2017. Disponível em: <<https://www.tecmint.com/create-a-linux-swap-file/>>. Citado na página 8.

Matt Vas. *How to clone disks with Linux dd command*. 2019. Disponível em: <<https://www.howtoforge.com/tutorial/linux-dd-command-clone-disk-practical-example/>>. Citado na página 10.

Microsoft. *tasklist | Microsoft Docs*. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/windows-server/administration/windows-commands/tasklist>>. Citado na página 6.

Nix Craft. *How To Check Swap Usage Size and Utilization in Linux - nixCraft*. 2019. Disponível em: <<https://www.cyberciti.biz/faq/linux-check-swap-usage-command/>>. Citado na página 8.

NixTutor. *Changing Priority on Linux Processes*. 2009. Disponível em: <<https://www.nixtutor.com/linux/changing-priority-on-linux-processes/>>. Citado na página 6.

OLIVEIRA, P. *Ajuda Linux - Dia 639 - Comando uname - Linux Solutions*. 2018. Disponível em: <<https://linuxsolutions.com.br/ajuda-linux-dia-639-comando-uname/>>. Citado nas páginas 5 e 13.

OLIVEIRA, S. *Descubra quanta energia seu PC consome e como reduzir isso*. 2019. Disponível em: <<https://canaltech.com.br/desktop/descubra-quanta-energia-seu-pc-consome-e-como-reduzir-isso/>>. Citado na página 15.

Oracle. *Introduction to the Scheduler - Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*. 2012. Disponível em: <https://docs.oracle.com/cd/E23824_01/html/821-1460/rmfss-2.html>. Citado na página 7.

Oracle. *Timesharing Scheduler (TS) (Installing and Administering Solaris Container Manager 1.1)*. 2012. Disponível em: <<https://docs.oracle.com/cd/E19112-01/ctr.mgr11/817-7551/fnzbi/index.html>>. Citado na página 7.

OS Tech Nix. *How To Suspend A Process And Resume It Later In Linux - OSTechNix*. 2017. Disponível em: <<https://www.ostechnix.com/suspend-process-resume-later-linux/>>. Citado nas páginas 6 e 7.

Prestashop. *Crie e desenvolva seu negócio com a PrestaShop*. 2019. Disponível em: <<https://www.prestashop.com/pt>>. Citado na página 3.

Prestashop. *System requirements for PrestaShop 1.7 :: PrestaShop Developer Documentation*. 2019. Disponível em: <<https://devdocs.prestashop.com/1.7/basics/installation/system-requirements/>>. Citado na página 14.

Project CSCI. *Scheduling in Linux by Chandrima Sarkar*. 2019. Disponível em: <https://www.cs.montana.edu/~chandrima.sarkar/AdvancedOS/CSCI560_Proj_main/index.html>. Citado na página 7.

Quora. *Why does Linux need a swap partition for RAM when Windows does not? If I add RAM then will I also need to repartition my Linux hard drive to match it each time?* - Quora. 2014. Disponível em: <<https://www.quora.com/Why-does-Linux-need-a-swap-partition-for-RAM-when-Windows-does-not-If-I-add-RAM-then-will-I-also-need-to-repartition-my-Linux-hard-drive-to-match-it-each-time>>. Citado na página 9.

SANTANA, P. *SystemInfo – Comando poderoso do Windows que muitos não conhecem!* / Paulo Santanna, MVP. 2012. Disponível em: <<https://paulosantanna.com/2012/11/06/systeminfo-comando-poderoso-do-windows-que-muitos-nao-conhecem/>>. Citado na página 6.

Shahriar Shovon. *Hex Editors on Linux – Linux Hint*. 2018. Disponível em: <https://linuxhint.com/hex_editor_linux/>. Citado na página 11.

Shapes Hed. *Linux and Unix ps command tutorial with examples* / George Ornbo. 2016. Disponível em: <<https://shapedhed.com/unix-ps/>>. Citado na página 7.

TecMundo. *Windows: como formatar e montar um novo disco rígido ou SSD* - TecMundo. 2013. Disponível em: <<https://www.tecmundo.com.br/disco-rigido/42432-windows-como-formatar-e-montar-um-novo-disco-rigido-ou-ssd.htm>>. Citado na página 10.

Tutorials Point. *pidof - Unix, Linux Command - Tutorialspoint*. 2019. Disponível em: <https://www.tutorialspoint.com/unix_commands/pidof.htm>. Citado na página 7.

WinDroidWiz. *[Trick]Pause/Resume ANY Task in Windows*. – WinDroidWiz. 2019. Disponível em: <<https://windroidwiz.wordpress.com/2016/07/30/trickpauseresume-any-task-in-windows/>>. Citado na página 7.

Wikipedia. *Scheduling (computing)*. 2019. Disponível em: <[https://en.wikipedia.org/wiki/Scheduling_\(computing\)#Windows](https://en.wikipedia.org/wiki/Scheduling_(computing)#Windows)>. Citado na página 7.