INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO - IFMT

CAMPUS CUIABÁ “OCTAYDE JORGE DA SILVA”

DEPARTAMENTO DE COMPUTAÇÃO

Gabriela Gomes dos Santos

Vitor Bruno de Oliveira Barth

COMPARAÇÃO ENTRE OS SISTEMAS OPERACIONAIS BASEADOS EM BSD, MINIX E BASEADOS EM GNU/LINUX NO CONTEXTO DA DISCIPLINA DE SISTEMAS OPERACIONAIS

Cuiabá

2017

Gabriela Gomes dos Santos

Vitor Bruno de Oliveira Barth

COMPARAÇÃO ENTRE OS SISTEMAS OPERACIONAIS BASEADOS EM BSD, MINIX E BASEADOS EM GNU/LINUX NO CONTEXTO DA DISCIPLINA DE SISTEMAS OPERACIONAIS

Relatório Técnico apresentado como requisito para aprovação da disciplina de Sistemas Operacionais, no Curso de Engenharia da Computação, no Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso.

Prof. Dr. Ed’Wilson Tavares Ferrreria

Cuiabá

2017

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc490504078)

[2 DESENVOLVIMENTO 3](#_Toc490504079)

[2.1 OBJETIVO GERAL 3](#_Toc490504080)

[2.1.1 Objetivos Específicos 3](#_Toc490504081)

[2.2 METODOLOGIA 3](#_Toc490504082)

[2.3 PROCEDIMENTOS EXPERIMENTAIS 6](#_Toc490504083)

[2.3.1 Testes com OpenBSD: 6](#_Toc490504084)

[2.3.2 Testes com MINIX: 10](#_Toc490504085)

[2.3.3 Testes com LINUX: 12](#_Toc490504086)

[2.4 RESULTADOS 14](#_Toc490504087)

[3 CONCLUSÕES E RECOMENDAÇÕES 15](#_Toc490504088)

[REFERÊNCIAS 16](#_Toc490504089)

# 1 INTRODUÇÃO

*Berkley Software Distribution* (BSD) é um Sistema Operacional baseado em Unix, que foi desenvolvido e distribuído pelo Grupo de Pesquisa em Sistemas de Computadores (CSRG) da Universidade da Califórnia, em Berkley, entre 1977 e 1995. Em 1995 foi lançada a última versão oficial do BSD. Logo após o lançamento desta versão, CSRG foi dissolvido e o desenvolvimento do BSD cessou. Desde então, diversas variantes, baseadas diretamente ou indiretamente no BSD, são mantidas pela comunidade [1].

Dentre essas variantes, destacam-se as distribuições de Código Aberto *FreeBSD, NetBSD* e *OpenBSD.* Elas utilizam um *kernel* monolítico, são gratuitas, e compartilham a maior parte de seu código em um único repositório central [1].

*Mini-Unix (*MINIX) é um Sistema Operacional de código aberto baseado em Unix, baseado em uma arquitetura de *microkernel.* Versões iniciais do MINIX foram criadas por Andrew S. Tanenbaum, lançadas em 1987, para propósitos educacionais, entretanto, a partir do MINIX 3, a meta de desenvolvimento primária foi alterada de um Sistema Operacional Educacional para um Sistema Operacional confiável baseado em um *microkernel* [2]*.*

*GNU/Linux* é um *kernel* modular*,* desenvolvido por Linus Torvalds em 1991. É considerado o maior projeto de código aberto já desenvolvido, e serve como base para inúmeras distribuições. Sua versão mais recente foi lançada em 15 de julho de 2017.

*Ubuntu* é um Sistema Operacional baseado no *kernel* *GNU/Linux.* É a distribuição com maior número de usuários domésticos, por possuir suporte a *hardwares* proprietários e interface amigável. Sua última versão foi lançada em 13 de abril de 2017, entretanto esta é considerada uma versão de testes. A última versão estável foi lançada em 21 de abril de 2016, e é chamada de 16.04 LTS. [4]

# 2 DESENVOLVIMENTO

2.1 OBJETIVO GERAL

* Verificar se Sistemas Operacionais Alternativos são capazes de suprir a necessidade de um Usuário Final Comum.

2.1.1 Objetivos Específicos

* Realizar testes de desempenho nos Sistemas Operacionais MINIX 3.3, *OpenBSD 6.1* e Ubuntu 16.04.
* Comparar o resultado obtidos nos testes de desempenho.

2.2 METODOLOGIA

Foi utilizado um computador com as seguintes especificações:

* Processador Intel i7 4810MQ de 2.8GHz a 3.8GHz, com 4 núcleos e 8 threads
* 12 GB de memória RAM DDR3 de 1333MHz
* SSD de 240GB, com taxa de transferência de dados de 600 Mbps
* Monitor com resolução de 1920x1080 pixels e taxa de atualização de 60 Hz
* Placa Ethernet Realtek RTL8168
* Placa Wifi Qualcomm Atheros AR9285

Foi realizada uma instalação limpa de cada um dos Sistemas Operacionais, utilizando os mesmos componentes durante todo o desenvolvimento deste trabalho.

O *boot* foi realizado em modo *Legacy*, com as opções *Fast Boot* e *Secure Boot* desativadas, para evitar incompatibilidade.

O computador estava em ambiente com temperatura controlada em 24ºC, em local arejado, sem obstrução nas saídas de ar para evitar gargalos de performance devido à superaquecimento.

Os testes foram divididos em duas categorias:

1. Testes de Usabilidade e Interação com o Usuário: analisam a adaptação do Sistema Operacional ao *hardware* utilizado, e se ele possui, nativamente, determinadas ferramentas consideradas essenciais para um Usuário Final.
2. Testes de Estresse: algoritmos que consomem, propositalmente, todos os recursos disponíveis, para se verificar como o Sistema Operacional se comporta sob situações de uso intenso.

Para os Testes de Usabilidade e Iteração com Usuário, foram levados em consideração:

* Experiência da Instalação: dificuldades encontradas durante a instalação, quantidade de tentativas até uma instalação bem-sucedida, e tempo.
* Experiência de Primeiro Uso: *softwares* instalados nativamente, resolução da tela, interface do usuário, conexão de rede *ethernet* e *wireless*
* Experiência de Produtividade: disponibilidade de *softwares* de escritório e navegadores de internet em seus repositórios, e quantidade de erros e travamentos durante 15 minutos de navegação na internet, onde acessou-se a rede social *Facebook,* o e-mail *Gmail*, e o site de vídeos *YouTube.*

Os seguintes testes de estresse foram feitos pelo *software stress-ng*. O *software stress-ng* exibe informações usando uma medida própria, chamada operações *bogus,* que indica quantas vezes o teste pôde ser executado no intervalo de tempo definido. Quanto maior, melhor. Durante os testes é verificado se há travamentos ou falhas. Cada teste é executado três vezes seguidas, com dois minutos de repouso entre cada uma das corridas, para evitar inconsistências.

Para cada um destes itens é executado um teste do *software stress-ng* que executa testes e determina algumas medidas relevantes para o *hardware* que se está medido. É também verificado se houve algum erro ou prejuízo à execução do Sistema Operacional.

* Performance de Memória Virtual:

Comando: *stress-ng --vm 1 --vm-bytes 90% --vm-method all --verify -t 2m –v --metrics*

Este teste cria um processo que tenta consumir 90% da memória virtual disponível no sistema.

Como resultado, foram avaliados o número de Operações *Bogus* por Segundo, indicando quantas operações *page in* e *page out* foram feitas por segundo.

* Performance de Memória RAM:

Comando: *stress-ng --brk 1 --verify -t 2m –v –metrics*

Este teste cria um processo que consome toda a memória RAM disponível, crescendo o segmento de dados uma página por vez. Caso um processo seja morto pelo Sistema Operacional por utilizar toda a memória, outro processo é criado automaticamente para substituí-lo.

Como resultado, foram avaliados o número de Operações *Bogus* por Segundo, que indica quantas operações *brk* (que servem para alocar uma página da memória) são realizadas por segundo.

* Performance de Disco

Comando: *stress-ng --hdd 10 --verify -t 2m –v –metrics*

Este teste cria dez processos que executam operações de escrita, leitura e deleção sequencial em disco. Foram escolhidos criar dez processos para verificar como o Sistema Operacional se comporta quando é necessário realizar acessos aleatórios em disco pois cada um dos processos opera sequencialmente, mas em uma posição diferente.

Como resultado, foram avaliados o número de Operações *Bogus* por Segundo, que indica quantas operações de escrita, leitura e deleção sequencial foram realizadas por segundo.

* Performance de Processamento

Comando: *stress-ng --cpu 128 --cpu-method matrixprod --verify -t 2m –v –metrics*

Este teste cria 128 processos que realizam multiplicação entre duas matrizes quadradas de ordem 128. Foi escolhido criar muitos processos para avaliar também como o Sistema Operacional lida com processamento paralelo e multiprocessamento extremo.

Como resultado, foram avaliados o número de Operações *Bogus* por Segundo, que indica vezes conseguiu-se realizar a multiplicação de tamanho 128x128 em um segundo.

## PROCEDIMENTOS EXPERIMENTAIS

2.3.1 Testes com OpenBSD 6.1:

Experiência de Instalação:

Foi utilizado a versão 6.1 deste Sistema Operacional (SO), disponível no site <<https://www.openbsd.org>>. Para a primeira tentativa de instalação, realizou-se o *download* da imagem ‘*install61.iso’*, entretanto essa imagem não foi reconhecida como “bootável” pelo aplicativo utilizado para copiar as imagens de instalação para o *pendrive* que serviria como mídia de *boot.* Foi necessário, então, realizar o *download* da imagem ‘*install61.fs’* que foi transferida para a mídia de instalação através do comando dd *if=install61.fs of=/dev/sdb1 bs=1m*, executado em um ambiente Ubuntu 16.04 LTS

1ª Tentativa de Instalação: O boot foi realizado sem nenhuma complicação, e foi exibida uma mensagem de Boas Vindas ao Programa de Instalação do *OpenBSD.* Verificou-se que Programa de Instalação é totalmente em Inglês, sem opção para escolher outro idioma.

Foram sugeridas as opções de Instalação Manual, Instalação Automática e Bash.

Foi escolhida a opção *AutoInstall.*

A primeira configuração proposta foi a escolha da interface de rede.

Escolheu-se a interface *athn0* que é a interface de rede *wireless.*

Após alguns testes de conexão, o Programa de Instalação exibiu erro e terminou abruptamente.

2ª Tentativa de Instalação: O computador foi conectado à rede local via cabo *ethernet* e então reiniciado por meio do comando *reboot.*

Escolheu-se a interface de rede *re0.*

Após alguns instantes, o Programa de Instalação exibiu erro e terminou abruptamente.

3ª Tentativa de Instalação: Reiniciou-se o computador por meio do comando *reboot.*

Desta vez, foi escolhida a opção *Install.*

A primeira opção foi a escolha de teclado. O *layout* de teclado do computador é *US International* de 101 Teclas, então escolheu-se a opção *us*.

Verificou-se que existe também a opção *br,* indicando que *openBSD* suporta o *layout* de teclado ABNT2.

Após escolhido o teclado, pediu-se que fosse inserido o nome do sistema. Foi dado o nome *testeSO.*

Apareceu, novamente, uma mensagem pedindo uma interface de rede. Verificou-se que havia uma nova interface, a *vlan0,* que foi a escolhida.

A configuração de rede não foi bem-sucedida, mas desta vez o instalador não terminou abruptamente, e sim pediu se deveria tentar reconfigurar a rede ou continuar sem rede.

Tentou-se, novamente, configurar a interface *re0*, e desta vez obteve-se sucesso.

Foi pedido o nome de domínio DNS, mas, por não haver um servidor DNS local, a entrada foi nula.

Pediu-se uma senha para o usuário *root.*

Perguntou-se se o serviço *sshd* deve ser iniciado por padrão. Foi permitido.

Perguntou-se se o Sistema de Janelas X deveria ser iniciado. Foi permitido.

Perguntou-se se iria ser configurado novo usuário. Foi recusado.

Perguntou-se se o usuário *root* poderia ser acessado por *ssh.* Foi permitido.

Perguntou-se o fuso horário, e foi sugerido *America/Cuiaba.* A sugestão foi aceita.

Perguntou-se qual o disco deveria ser o padrão. Somente um foi encontrado, o disco *sd0,* que era a própria mídia de instalação.

A instalação foi cancelada pelo usuário através do comando *Ctrl+C.*

Esta tentativa e o tempo gasto nela não foram contabilizados no quadro geral, pois como verificou-se na 4ª Tentativa, o erro foi causado devido a falha na configuração do computador de testes por parte dos autores.

4ª Tentativa: O computador foi reiniciado para que fossem conferidas as configurações de disco da BIOS. Viu-se que o modo SATA escolhido era RAID. Suspeitou-se que esta era a causa da falha de instalação. O modo SATA foi colocado em AHCI, e o computador foi reiniciado.

Foram executados os mesmos procedimentos da 3ª tentativa, exceto a tentativa de configuração da interface de rede *vlan0,* que foi ignorada.

Perguntou-se qual o disco deveria ser o padrão. Foram encontrados dois discos: *sd0,* que era o SSD de 240GB, e *sd1,* que era a mídia de instalação. Foi escolhido o disco *sd0.*

Perguntou-se qual o sistema de partições deveria ser utilizado. Escolheu-se o padrão MBR.

Perguntou-se qual o *layout* de partições deveria ser utilizado. Foi escolhido o *layout* automático, oferecido pelo Programa de Instalação.

A instalação se iniciou.

Em 13 segundos, a instalação havia sido concluída, sem erros.

Perguntou-se a localização dos *sets.* Escolheu-se o modo http, e entrou-se com o espelho localizado na Universidade Federal do Paraná, cujo link é << https://openbsd.c3sl.ufpr.br/>>.

Todos os *sets* disponíveis foram baixados, verificados, e instalados normalmente.

A instalação foi terminada e foi mostrada uma tela de Felicitações devido à instalação bem-sucedida do OpenBSD. Desde a conclusão do Download do Sistema Operacional até a tela, a instalação demorou 1 hora e 12 minutos. O computador foi então reiniciado sem a mídia de instalação, para que então pudessem começar os testes.

Experiência de Primeiro Uso:

Ao reiniciar, o Sistema de Janelas X foi iniciado diretamente. Não havia nenhum *software* gráficoinstalado por padrão. A resolução da tela estava em seu máximo, 1920x1080, e não foi verificado nenhum erro, travamento ou falha.

O acesso à internet foi testado através do comando *ping google.com.* Quando o cabo de rede estava presente, o servidor do Google pode ser acessado. Entretanto, como não há interface, a rede *wireless* deveria ser configurada através do console.

Foi então pesquisado como é feita a conexão em rede sem fio no Sistema Operacional *openBSD.* Encontrou-se no site do *openBSD* um tutorial de configuração de rede *wireless.* Para tal, era necessário editar o arquivo */etc/hostname.athn0* (sendo *athn0* a interface de rede qual se desejava configurar)*.* O único editor de texto presente no Sistema era o Vi. Os autores deste relatório não possuíam experiência para utilizar o Vi. Por não possuir controle da placa de som, cada comando incorreto causava um apito do *buzzer* de emergência da placa mãe. Por este motivo, decidiu-se instalar o editor de textos *nano.* Para instalar o *nano,* executou-se o comando *pkg\_add –v* *https://openbsd.c3sl.ufpr.br/pub/OpenBSD/6.1/packages/amd64/nano-2.7.5.tgz.* O editor de texto *nano* foi instalado com sucesso.

Para tentar configurar a rede *wireless,* o arquivo */etc/hostname.athn0* foi editado, deixando-o como segue:

*nwid nome-da-rede*

*wpakey senha-da-rede*

*dhcp*

O SO foi então reiniciado sem o cabo de rede. Verificou-se através do comando *ping google.com* que o computador estava, agora, conectado à rede *wireless.* Terminando a configuração inicial, deu-se início aos testes de performance.

Testes de Performance:

Instalou-se a ferramenta *stress-ng* v0.08.10, compilando-a a partir de seu código fonte. Para isso, executou-se os seguintes comandos:

*pkg\_add wget*

*wget http://kernel.ubuntu.com/~cking/tarballs/stress-ng/stress-ng-0.08.10.tar.gz*

*pkg\_add gmake*

*CC=gcc gmake install*

Teste de Performance de Memória Virtual: O sistema sofreu de *slowdowns* e travamentos durante a execução dos testes. Após a finalização deste teste, o sistema voltou ao seu estado normal. O uso de processador manteve-se estável em menos de 2% (±1%) durante a realização dos testes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 45.402,27 | 45.429,47 | 45.715,38 | 45.515,70 |

Teste de Performance de Memória RAM: O sistema travou completamente, sendo inutilizável durante a execução dos testes. Após a finalização deste teste, o sistema voltou ao seu estado normal. O uso de processador manteve-se estável em pouco mais de 2% (±1%) durante a realização dos testes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 239.660,66 | 234.998,49 | 239.961,33 | 238.206,82 |

Teste de Performance de Disco: O sistema continuou totalmente utilizável durante a execução dos testes. O uso de manteve-se estável em 11% (±1%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 5.395,01 | 5.253,00 | 5.260,84 | 5.302,95 |

Teste de Performance de Processamento: O sistema continuou totalmente utilizável durante a execução dos testes. O uso de processador manteve-se estável em 112% (±20%)[[1]](#footnote-1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 90,19 | 90,78 | 89,78 | 90,25 |

Experiência de Produtividade:

Verificou-se que o SO *openBSD* possui em seus repositórios o *software LibreOffice* v5.2.4, que pode ser instalado através do comando *pkg\_add libreoffice*. Como teste, este relatório técnico foi aberto no *software LibreOffice Writer* e esta linha foi escrita utilizando o sistema operacional *openBSD.*

Para teste de navegação de internet, instalou-se o navegador *Mozilla Firefox 52.0.2* presente no repositório padrão, e instalável através do comando *pkg\_add firefox.* Todas as tarefas propostas (acessar o *Facebook, Gmail* e *YouTube*)foram realizadas sem problemas, entretanto, por não haver saída de som, o funcionamento do *YouTube* foi comprometido.

2.3.2 Testes com MINIX 3.3:

Experiência de Instalação:

Foi utilizado a versão 3.3.0 deste Sistema Operacional (SO), disponível no site <<https://www.minix3.org>>. O Sistema Operacional MINIX3 não possui suporte para instalação via USB, então foram necessários alguns procedimentos [3]:

1. Foi copiada a imagem do SO Ubuntu 16.04 para um p*endrive* através do *software Unetbootin*
2. Foi copiada o arquivo de imagem do SO MINIX para um outro p*endrive*
3. Realizou-se *boot* numa imagem *live* do SO *Ubuntu* 16.04
4. Foi instalado o *software qemu* através do comando *sudo apt-get install qemu-system-x86*
5. Foi executado o comando *sudo qemu-system-i386 -m 512 -drive file=/dev/sda,format=raw,media=disk -cdrom /media/ubuntu/pendrive/MINIX3.iso -boot d &*

Então foi iniciada uma máquina virtual com o instalador do MINIX, sobre o Sistema Operacional Ubuntu 16.04 rodando no Pendrive, contudo todas as alterações feitas pelo instalador iriam para o disco /dev/sda. (15h20)

1ª Tentativa de Instalação: Escolheu-se a opção *1 Regular MINIX*. Foi feito *login* com usuário *root,* sem senha e executou-se o comando *setup* para que a instalação se iniciasse.

Apareceu uma mensagem de bem-vindo ao instalado do MINIX 3 e que caso apareçam ‘dois pontos’ (:) deve ser apertada a tecla Enter para continuar. Não foi proposto mudança de idioma, indicando que a instalação seria toda em ingles

Foi pedido que se selecionasse qual o padrão de teclado a ser utilizado. Escolheu-se *us-std,* que corresponde ao teclado padrão do computador de testes. Verificou-se que a opção *abnt2* estava presente, indicando que MINIX3 é compatível com o teclado brasileiro.

Pediu-se o modo de instalação, automático ou *expert.* Escolheu-se automático.

Perguntou-se qual o disco a ser utilizado. Escolheu-se o disco 0, que correspondia ao SSD.

Perguntou-se qual o tamanho desejado da partição /home. Deixou-se o padrão (45653 MB).

Perguntou-se o tamanho do bloco. Escolheu-se o padrão (4kB).

Então se deu início à formatação do disco e instalação do sistema.

Perguntou-se qual a placa de rede utilizada. Escolheu-se a que estava indicada (Intel PRO/1000 Gigabit), e indicou-se que a configuração de rede seria feita por DHCP.

Reiniciou-se o sistema.

O tempo total de instalação foi de 15 minutos, contando a partir do momento que a mídia de *boot* foi feita. Entretanto, caso seja contado desde o momento do download da imagem, o tempo total aumenta para 2 horas e 37 minutos, devido ao tempo de pesquisa sobre como realizar a instalação via USB.

O computador foi então reiniciado sem a mídia de instalação, para que então pudessem começar os testes.

Experiência de Primeiro Uso: Logo no primeiro *boot,* foi apresentada uma mensagem de *kernel panic*, causada pelo driver *at\_wini,*que impedia o acesso o *boot* do sistema operacional*.* Após pesquisas, descobriu-se que esse erro é causado pelo *driver* de comunicação com disco. O MINIX 3, por padrão, aceita discos pelo padrão IDE, entretanto neste computador está configurado o padrão AHCI. Por isso, foi necessário alterar a opção de *boot,* adicionando a opção *ahci=yes* na linha de início do sistema.

No segundo *boot* continuou sendo exibido mensagem de *kernel panic,* agora causada pelo *driver* de rede. Para trocar o *driver* de rede, executou-se o comando *netconf* e escolheu-se a opção *Realtek 8139 Based Card,* que não estava disponível durante o *setup* e corresponde exatamente ao modelo de placa de rede presente no computador.

O sistema foi então reiniciado e foi exibida uma mensagem de Bem-Vindo ao MINIX3. Não havia interface gráfica instalada.

Realizou-se o teste de rede com o comando *ping google.com*, e verificou-se que o Sistema não estava conectado à rede. Após várias tentativas de configuração do *driver* de rede, foi decidido colocar o SSD em um computador com placa de rede genérica suportada pelo sistema, para que fosse realizada a instalação dos *softwares* de teste.

Testes de Performance:

Para se instalar o *software* de testes *stress-ng* e as ferramentas de compilação*,* executou-se os seguintes comandos, o SSD foi colocado em um outro computador, cuja placa de rede era compatível com MINIX, e se realizou os seguintes comandos.

*pkgin install gmake*

*pkgin install clang*

*pkgin install git-base*

*pkgin instal binutils*

*git clone git://github.com/ColinIanKing/stress-ng*

*cd stress-ng*

*CC=clang gmake install*

Então o SSD foi colocado no computador inicial para que fossem executados os testes.

Teste de Performance de Memória Virtual: O sistema não sofreu de travamentos durante a execução dos testes, e o uso do processador manteve-se estável em pouco mais de 3% (±1%). Entretanto, o resultado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 0 | 0 | 0 | 0 |

Teste de Performance de Memória RAM: O sistema matou repetidamente os processos criados por falta de memória, causando falha de segmentação. O resultado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 0 | 0 | 0 | 0 |

Teste de Performance de Disco: O sistema matou repetidamente os processos criados alegando não haver espaço suficiente em disco. O resultado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 0 | 0 | 0 | 0 |

Teste de Performance de Processamento: O sistema permaneceu estável durante a execução dos testes. O uso do processador manteve-se em torno de 22% (±2%). Entretanto, o resultado retornado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 0 | 0 | 0 | 0 |

Experiência de Produtividade:

Por não possuir uma interface gráfica facilmente configurável e nem conexão à internet estável, o sistema MINIX os testes de produtividade não puderam ser realizados no sistema operacional MINIX.

2.3.3 Testes com Ubuntu 16.04:

Experiência de Instalação:

Foi utilizado a versão 16.04 LTS deste Sistema Operacional (SO), disponível no site <<https://www.ubuntu.com>>. A imagem foi copiada para um pendrive através do *software Unetbootin*

Realizou-se *boot* normalmente. Uma interface gráfica de instalação foi aberta, e perguntou-se o idioma a ser utilizado. Escolheu-se português. Foi também perguntado o modelo de teclado, e localização. Foram todas escolhidas de acordo com a configuração do computador.

O tempo total de instalação foi de 7 minutos, contando a partir do momento do download da imagem.

O computador foi então reiniciado sem a mídia de instalação, para que então pudessem começar os testes.

Experiência de Primeiro Uso: O primeiro *boot* ocorreu normalmente. O cabo de rede foi reconhecido e a conexão via *wireless* era possível através de uma interface. A resolução definida era a máxima suportada pelo monitor.

Deu-se início aos testes de performance.

Testes de Performance:

Para se instalar o *software* de testes *stress-ng* executou-se os seguintes comandos:

*sudo apt-get install git build-essential*

*git clone git://github.com/ColinIanKing/stress-ng*

*CC=clang gmake install*

Teste de Performance de Memória Virtual: O sistema não sofreu de travamentos durante a execução dos testes, e o uso do processador manteve-se estável em menos de 2% (±1%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 91.451,20 | 91.388,10 | 91.368,27 | 91.402,52 |

Teste de Performance de Memória RAM: O sistema ficou travado durante a maior parte do tempo de execução dos testes. O uso de processador se manteve estável torno de 3% (±1%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 287.052,11 | 290.518,82 | 290.622,20 | 289.397,71 |

Teste de Performance de Disco: O sistema matou repetidamente os processos criados alegando não haver espaço suficiente em disco. O resultado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 5.142.55 | 5.145,93 | 5.488,09 | 5.258,85 |

Teste de Performance de Processamento: O sistema permaneceu estável durante a execução dos testes. O uso do processador manteve-se constantemente em 100% (±2%). Entretanto, o resultado retornado foi nulo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parâmetro | Rodada 1 (R1) | Rodada 2 (R2) | Rodada 3 (R3) | Média  (R1+R2+R3)/3 |
| Operações *Bogus* por Segundo | 91,16 | 85,30 | 84,40 | 86,95 |

Experiência de Produtividade:

O sistema operacional Ubuntu 16.04 possui, nativamente, a suíte de aplicativos de escritório *LibreOffice 5.2.4* e o navegador *web* *Mozilla Firefoz 48.0.2.*

Foram realizados testes de navegação na *internet. Como* Ubuntu 16.04 possui, nativamente, suporte à áudio, todos os testes propostos foram bem-sucedidos.

2.4 RESULTADOS

|  |  |  |  |
| --- | --- | --- | --- |
| **TESTES DE USABILIDADE** | | | |
| **I – Experiência de Instalação** | | | |
| **Atributo** | **OpenBSD 6.1** | **MINIX 3.3** | **UBUNTU 16.04** |
| Tentativas de Instalação | 3 | 1 | 1 |
| Tempo Total de Instalação (Contabilizando todas as Tentativas e Pesquisas Necessárias) | 1 hora e 12 minutos | 2 horas e 37 minutos | 7 minutos |
| Idioma da Instalação | Inglês | Inglês | Português |
| Interface de Instalação | Console | Console | Gráfica |
| **II – Experiência de Primeiro Uso** | | | |
| **Atributo** | **OpenBSD 6.1** | **MINIX 3.3** | **UBUNTU 16.04** |
| Resolução de Tela | 1920x1080 | 1024x768 | 1920x1080 |
| Erro ou Travamento | Não | 2 *kernel panics* | Não |
| Interface Gráfica Padrão | Presente, mas confusa e deselegante. | Console | Presente e com todas as configurações esperadas de uma interface |
| *Softwares* de Escritório Presentes | Nenhum | Nenhum | Sim |
| Conexão de Rede via cabo *ethernet* | Sim | Não | Sim |
| Conexão de Rede via *wireless* | Sim, após configurações via console | Não há suporte. | Sim, nativamente |
| **III – Testes de Performance** | | | |
| **Atributo** | **OpenBSD 6.1** | **MINIX 3.3** | **UBUNTU 16.04** |
| Performance de Memória Virtual  (*Bogus* Ops/s) | 45.515,70 | 0 | 91.402,52 |
| Performance de Memória RAM  (*Bogus* Ops/s) | 238.206,82 | 0 | 289.397,71 |
| Performance de Disco  (*Bogus* Ops/s) | 5.302,95 | 0 | 5.258,85 |
| Performance de Processamento  (*Bogus* Ops/s) | 90,25 | 0 | 86,95 |
| **IV – Experiência de Produtividade** | | | |
| **Atributo** | **OpenBSD 6.1** | **MINIX 3.3** | **UBUNTU 16.04** |
| Aplicativos de Escritório | *LibreOffice 5.2.4* presente em seu repositório padrão. | Não foi possível avaliar | *LibreOffice 5.2.4* instalado por padrão |
| Navegador de Internet | *Mozilla Firefox 52.0.2* presente em seu repositório padrão. | Não foi possível avaliar | *Mozilla Firefox 48.0.2* instalado por padrão |
| Navegação na Internet | 75% | 0% | 100% |

3 CONCLUSÕES E RECOMENDAÇÕES

Através das experiências de instalação, primeiro uso e produtividade pode-se verificar que, dos Sistemas Operacionais avaliados somente o Ubuntu 16.04 é amigável ao usuário final.

*openBSD* e *MINIX* mostraram-se difíceis de se instalar, por realizarem perguntas específicas que o usuário comum poderia não saber responder, e também difíceis de se utilizar para pesquisa ou trabalho, por não ter suporte à *hardwares* essenciais, como áudio e rede sem fio, nativamente.

Para trabalhos futuros, pretende-se escolher outro *software* de testes sintéticos, visto que *MINIX* não foi compatível com *stress-ng*, e também realizar estes testes em outros *hardwares,* para verificar o comportamento destes Sistemas Operacionais em outros ambientes.

REFERÊNCIAS

[1] WIKIPÉDIA. *Berkley Software Distribution.* Disponível em: <<https://en.wikipedia.org/wiki/Berkeley\_Software\_Distribution>>. Acesso em: 11 ago. 17.

[2] \_\_\_\_\_\_\_\_\_. *MINIX.* Disponível em: << https://en.wikipedia.org/wiki/MINIX >>. Acesso em: 11 ago. 17.

[3] MINIX3 WIKI. *Installing MINIX 3 using USB.* Disponível em: << http://wiki.minix3.org/doku.php?id=usersguide:x86usbworkaround>> Acesso em: 11 ago. 17.

[2] \_\_\_\_\_\_\_\_\_. *Ubuntu.* Disponível em: << https://en.wikipedia.org/wiki/Ubuntu >>. Acesso em: 11 ago. 17.

1. O processado utilizado possui a tecnologia Intel® *Turbo Boost* 2.0 que aumenta a frequência do *clock* do processador durante os picos de carga. O SO *openBSD* realiza os cálculos de uso do processador sobre o *clock* padrão, logo gera um valor maior que 100% por um curto período de tempo, quando o processador realiza este balanceamento. A taxa de variação desta medida torna-se alta pois, quando o processador está em alta temperatura, a frequência do *clock* retorna ao padrão para evitar danos ao *hardware.* [↑](#footnote-ref-1)