

Sirley Maria Alberton

Aprendendo LINUX Através de Atividades Práticas

Monografia de Pós-Graduação “*Lato Sensu*”
apresentada ao Departamento de Ciência da
Computação para obtenção do título de Especialista
em “Administração em Redes Linux”.

Orientadora

Profa. Kátia Cilene Amaral Uchôa

Co-orientador

Prof. Anderson Gomes

LAVRAS

MINAS GERAIS – BRASIL

2005

Aprendendo LINUX Através de Atividades Práticas

Monografia de Pós-Graduação “*Lato Sensu*”
apresentada ao Departamento de Ciência da
Computação para obtenção do título de Especialista
em “Administração em Redes Linux”.

APROVADA em ____ de _____ de _____

Prof. _____

Prof. _____

Profa. Kátia Cilene

(Orientadora)

LAVRAS

MINAS GERAIS – BRASIL

Agradecimentos

Os meus agradecimentos são aos meus pais, Florida Pusch e Germano Alberton, que apesar de toda dificuldade não me deixaram sem o ensino básico. Ensino este essencial para que eu pudesse hoje estar cursando esta pós-graduação.

Em especial a minha mãe que com exemplo de vida me induziu a busca pelo conhecimento e aperfeiçoamento. Seu anseio pelo saber foi minha inspiração para chegar ao final desta jornada.

Agradeço a orientadora, Profa. Kátia Cilene Amaral Uchôa, que com paciência e dedicação me auxiliou de forma expressiva neste árduo trabalho.

Estendo meus agradecimentos a todos que estiveram ao meu lado, meus irmãos Vera e Edson, aos meus sete sobrinhos e aos colegas do curso e do trabalho que de uma forma ou de outra sempre me ofereceram apoio.

Resumo

O trabalho apresentado nesta monografia é uma proposta para aprendizado do Linux utilizando tarefas para aplicação prática, dividido em três fases: Ambientação; básico e intermediário. Esta proposta foi desenvolvida para ambientação em Linux de um pequeno grupo de pessoas com perfil técnico e administrativo do setor onde a autora da monografia trabalha. Mesmo assim este trabalho pode ser adequado para utilização por outros grupos de interesse, desde que sejam iniciantes no sistema operacional Linux. Trata-se de uma seqüência de atividades com tarefas cuidadosamente elaboradas para serem executadas diretamente no Linux. O aprendizado do Linux utilizando esta proposta é construído pela experiência efetiva na execução das tarefas.

Sumário

1	INTRODUÇÃO.....	7
2	UTILIZAÇÃO DAS TAREFAS PARA APRENDIZADO.....	11
3	METODOLOGIA UTILIZADA PARA CRIAÇÃO DAS TAREFAS.....	14
3.1	USO DO MODO TEXTO.....	15
3.2	ESCOLHA DOS COMANDOS	16
3.3	CRIAÇÃO DAS TAREFAS.....	34
3.3.1	<i>Processo de Criação das Tarefas</i>	<i>36</i>
3.4	DIVISÃO DAS FASES	41
4	APLICAÇÃO E RESULTADOS	43
5	CONCLUSÃO.....	47
6	REFERÊNCIAS BIBLIOGRÁFICAS	49
7	APÊNDICE A – QUESTIONÁRIO PERFIL.....	52
8	APÊNDICE B – QUESTIONÁRIO AVALIAÇÃO.....	53
9	APÊNDICE C – TAREFAS.....	54
9.1	- PRIMEIRA FASE – AMBIENTAÇÃO	54
9.1.1	<i>Conceitos básicos</i>	<i>54</i>
9.1.2	<i>Efetando logon/logoff.....</i>	<i>55</i>
9.1.3	<i>Desligando e Reiniciando.....</i>	<i>57</i>
9.1.4	<i>Primeiros comandos.....</i>	<i>59</i>
9.1.5	<i>Comando clear.....</i>	<i>61</i>
9.1.6	<i>Árvore de diretórios.....</i>	<i>62</i>
9.1.7	<i>Características do Linux</i>	<i>64</i>
9.2	SEGUNDA FASE – BÁSICO	66
9.2.1	<i>Como Obter Ajuda.....</i>	<i>66</i>
9.2.2	<i>Manipulação de Arquivos e Diretórios</i>	<i>70</i>
9.2.3	<i>Permissões de acesso a arquivos e diretórios</i>	<i>74</i>
9.2.4	<i>Criando Atalhos</i>	<i>77</i>
9.2.5	<i>Editores de texto</i>	<i>81</i>
9.2.6	<i>Utilitários de texto</i>	<i>85</i>
9.2.7	<i>Compressão e empacotamento de arquivos e diretórios</i>	<i>87</i>
9.2.8	<i>Data e Hora</i>	<i>88</i>
9.3	TERCEIRA FASE – INTERMEDIÁRIO	90
9.3.1	<i>Administração de usuários</i>	<i>90</i>
9.3.2	<i>Gerenciamento de processos.....</i>	<i>94</i>

9.3.3	<i>Gerenciamento do sistema</i>	96
9.3.4	<i>Criando Apelidos para Comandos (Alias)</i>	97
9.3.5	<i>Agendamento de Tarefas</i>	100
9.3.6	<i>Shell Script</i>	101
9.3.7	<i>Equivalências do Windows/DOS e Linux</i>	104
10	APÊNDICE D – QUESTIONÁRIOS RESPONDIDOS	107
10.1	PRIMEIRO VOLUNTÁRIO: ISAAC LUIZ SARAH SIDOU	107
10.2	SEGUNDO VOLUNTÁRIO: MARCELO CRUZ PRAUDE	110
10.3	TERCEIRO VOLUNTÁRIO: REINALDO ANTONIO VALÉRIO.....	113
10.4	QUARTO VOLUNTÁRIO: RUBENS CRUZ PRAUDE.....	116

1 Introdução

O sistema operacional Linux e o *software* livre vêm ganhando espaço cada vez mais significativo junto aos profissionais de TI. O Linux hoje é uma alternativa viável, de qualidade e desempenho comparado a qualquer sistema operacional comercial. Isto se deve a sua portabilidade, estabilidade, segurança e relação custo - benefício. O *software* livre por sua vez é alternativa viável ao modelo atual de licenciamento de *software*.

O GNU/Linux surgiu pelos esforços do revolucionário Richard M. Stallman que não se conformou com as restrições impostas pelo *software* proprietário e idealizou o projeto GNU (criação de um sistema operacional completo e totalmente livre). O projeto GNU defende a total liberdade dos usuários em executar, copiar, distribuir, avaliar, modificar e aperfeiçoar o código fonte. A junção do projeto GNU ao Linux (*Kernel*) criado por Linus Torvalds, formou o sistema operacional GNU/Linux. Mais detalhes sobre o projeto GNU pode ser encontrado nos artigos de Richard M. Stallman disponíveis em (STALLMAN, 2000) e (STALLMAN, 2001).

O Governo Federal Brasileiro define, entre as diretrizes¹ e princípios para gestão de Tecnologia e Informação, a utilização de *software* livre como opção estratégica. Esta opção reflete o interesse do Governo em promover a independência tecnológica e de fornecedores, racionalizar os recursos aplicados sem comprometer a qualidade dos serviços prestados.

¹ Diretrizes: <http://www.softwarelivre.gov.br/diretrizes/>

Para auxílio na migração dos setores públicos o Governo Federal disponibiliza os referenciais, Guia Livre e *e-Ping*. O Guia Livre visa ser um referencial de migração para *software* livre criado pelo Governo Federal (SLTI/ITI, CISL, CTSSL, 2004) e o *e-Ping* (GOVERNO FEDERAL, 2005) define um conjunto de padrões de interoperabilidade de Governo Eletrônico.

Com esta mobilização do Governo para utilização de *software* livre nos órgãos públicos, existe uma sinalização para que isto ocorra também em órgãos financeiros como a Caixa Econômica Federal, empresa na qual trabalha a autora desta monografia na função de analista.

Para ambientação, neste novo sistema operacional, dos funcionários do setor onde a autora desta monografia trabalha, foi desenvolvida uma proposta que visa facilitar o aprendizado do Linux, através de atividades práticas. A presente monografia apresenta esta proposta, a fim de que possa ser um fio condutor no desenvolvimento de trabalho desta natureza.

O objetivo deste trabalho é apresentar um direcionamento para o aprendizado do Linux, com o desenvolvimento de atividades para aplicação prática, direcionadas para um grupo específico de pessoas (profissionais da área de TI e administrativo sem nenhum conhecimento do Linux). Estas atividades estão divididas em fases de aprendizado e colocadas em ordem crescente de dificuldade. Este direcionamento não tem o objetivo constituir um manual do Linux e sim servir para indicar o que deve ser estudado apresentando em forma de atividades práticas.

Mesmo tendo sido criadas para um grupo específico de pessoas, com um determinado perfil, as atividades podem ser utilizadas por outros grupos com outras características, como por autodidatas ou em grupos de estudo sistematizado ou até mesmo para complementar treinamento formal. Isto é

possível uma vez que as atividades contemplam o básico para que se conheça o Linux, sendo uma introdução ao mundo deste sistema operacional, sem especificação de nenhum aplicativo proprietário. Com pouca ou nenhuma adequação outros grupos podem se beneficiar deste trabalho, desde que sejam iniciantes em Linux.

As atividades foram desenvolvidas com base em conteúdos encontrados em livros, Internet e ajuda do próprio Linux, também foram baseadas no conhecimento empírico da autora em repassar orientação técnica. Nessa proposta de direcionamento de aprendizado do Linux foi levada em consideração a necessidade de desenvolver o espírito de pesquisa, essencial para a formação de um bom profissional de TI.

Posteriormente, todas as atividades foram efetivamente testadas em três distribuições do Linux e ainda experienciadas por uma amostra pequena de pessoas que puderam melhor validar os resultados da utilização das mesmas para aprendizado do Linux.

Para validação desta proposta, as pessoas que utilizaram efetivamente as atividades foram submetidas a um questionário. Os questionários podem ser encontrados nos Apêndices A e B e os questionários respondidos pelo grupo que utilizou as tarefas para aprendizado do Linux estão no Apêndice D. Os testes das tarefas e validação dos resultados são tratados em detalhes no Capítulo 4 desta monografia.

As distribuições escolhidas para testar as atividades foram: Mandrake, SUSE e Debian. A distribuição Mandrake foi utilizada por ser a única, a princípio, instalada na estação disponível para teste. As distribuições SUSE e Debian foram escolhidas pelo fato de existir uma indicação por parte da empresa onde a autora da monografia trabalha em utilizá-las. Mesmo assim, o trabalho

pode ser aplicado em outras distribuições, uma vez que foram selecionados apenas os comandos básicos. Com raras exceções todos estes comandos devem estar presentes em quase todas as distribuições Linux.

As atividades estão divididas em fases distintas. Estas fases compreendem: Ambientação; básico; intermediário; avançado e experiente. Nesta monografia serão tratadas apenas as três fases iniciais por estarem direcionadas para os iniciantes em Linux, as demais fases mais direcionadas aos profissionais experientes da área, serão tratadas em trabalho futuro. Também as três primeiras fases exigem apenas uma estação com Linux para que possam ser executadas, enquanto para a realização das demais fases, avançado e experiente, será necessário um ambiente mais elaborado com servidores e estações em rede.

Assim, portanto, o contexto deste trabalho encontra-se organizado da seguinte forma: O Capítulo 2 apresenta a utilização das tarefas para aprendizado; o Capítulo 3 apresenta a metodologia utilizada para criação das tarefas; o Capítulo 4 mostra como foram feitos os testes das atividades no Linux e apresenta os resultados da efetiva utilização das mesmas por um pequeno grupo de pessoas; o Capítulo 5 apresenta a conclusão geral do trabalho. Além disso, existem quatro Apêndices: O Apêndice A contém o questionário para levantamento do perfil do aprendiz; enquanto no Apêndice B está o questionário para avaliação das atividades para aprendizado do Linux; O Apêndice C está dividido em três etapas: Etapa I apresenta as atividades da fase I, ambientação; etapa II, contém as atividades da fase II, básico, enfim a etapa III apresenta as atividades da fase III, intermediário, e finalmente no Apêndice D, estão os questionários respondidos pelos voluntários que utilizaram as tarefas para aprendizado do Linux.

2 Utilização das Tarefas para Aprendizado

A decisão de utilizar tarefas como forma para o aprendizado do Linux foi tomada porque com esta ferramenta existe a possibilidade de uma indicação do que estudar. Desta forma facilita o aprendizado do Linux, porque além de proporcionar uma equalização do conhecimento entre os membros da equipe propicia experiências reais na execução dos comandos e incentiva a pesquisa.

Para incentivar a pesquisa as tarefas foram construídas de forma que não apresentam toda a informação necessária para que sejam executadas, havendo sempre necessidade de pesquisa adicional. Portanto, as tarefas podem ser consideradas um direcionamento para estudo e não um manual do Linux.

A proposta para aprendizagem do Linux utilizando tarefas apresenta o conteúdo dividido em fases de aprendizado e numa seqüência lógica e crescente de dificuldade. As divisões das fases propostas neste trabalho são:

- **Ambientação** – Conhecimento inicial do sistema. Como exemplo o *logon/logoff*, reiniciar/desligar, interfaces texto/gráfica, estrutura de diretórios e comandos básicos como `pwd`, `cd` e `ls`.

- **Básico** – Comandos mais utilizados para as tarefas do dia-a-dia, como os para manipulação de arquivos: `cp`, `tar` e `mv`.

- **Intermediário** – Comandos e configurações mais complexas como administração de usuários, administração do sistema, agendamento de tarefas, criação de *Shell Script* entre outras.

As tarefas para as três fases indicadas acima estão disponíveis no Apêndice C desta monografia.

Para utilização das tarefas o interessado deve observar a necessidade de alguns pré-requisitos, que são:

- Acesso a material para pesquisa;
- Ter acesso a uma estação com Linux instalado, com uma das distribuições indicadas neste trabalho (Mandrake, SUSE ou Debian,) ou outra qualquer, desde que se assuma o risco de algum comando não funcionar como previsto na atividade;
- Possuir acesso a senha do *root* desta estação;
- Estar cadastrado na estação como usuário comum.

O aprendizado do Linux através do uso das tarefas será construído pela experiência na efetiva execução da tarefa proposta e cada experiência servirá também para execução das tarefas seguintes. Existe um elo que interliga uma atividade com as outras, de forma que o conhecimento adquirido ao finalizar uma atividade será aproveitado nas atividades seguintes. A isto se chama recontextualização, que significa “conectar novos conceitos ou habilidades com as experiências passadas” conforme (BEN-HUR, 2005). A recontextualização é um dos princípios da reconstrução cognitiva que foi influenciada pela teoria do construtivismo como descrito em (BEN-HUR, 2005), (UCHÔA, 2001) e (SASTRE, 2002).

De certa forma, as práticas que privilegiam essa proposta estão de acordo com o construtivismo. Construtivismo é uma teoria que diz que o indivíduo tem papel ativo na construção dos conhecimentos novos. Segundo esta teoria, o verdadeiro conhecimento é aquele que é fruto de uma elaboração pessoal, que resulta de um processo de construção feita pelo indivíduo baseado em experiências anteriores. Esta é uma teoria psicológica, originalmente devida a

Jean Piaget como descrito em (BEN-HUR, 2005) e (SASTRE, 2002) e reforçada por autores construtivistas como Lev Semenovitch Vygotsky como apresentado em (UCHÔA, 2001).

As pesquisas para criação das tarefas foram baseadas em livros e material encontrado na Internet sobre os comandos do Linux, sendo os principais: O livreto “Guia de Consulta Rápida – Comandos do Linux”, de Roberto G. A. Veiga (VEIGA, 2004). Este livreto apresenta uma grande quantidade de comandos com detalhes sobre suas opções; outro material também de igual valor é o livro “Comandos do Linux – Uso eficiente e avançado” de Jário Araújo (ARAUJO, 2001). Também apresenta uma grande quantidade de comandos com detalhes sobre suas funcionalidades e opções; o material encontrado na Internet, o “Guia Foca GNU/Linux”² (SILVA, 2005), produzido por Gleydson Mazioli Silva, apresenta os tópicos com riqueza de detalhes e divide em fases de dificuldade, iniciante, intermediário e avançado. Sendo uma ótima fonte de pesquisa; o livro “Linux – Administração e Suporte”, de Chuck V. Tibet (TIBET, 2001), apesar de estar mais direcionado para usuários mais experientes, também apresenta um bom material para iniciante. Os demais livros relacionados nas referências bibliográficas tiveram contribuição em menor escala, devido sua abordagem privilegiar configurações mais complexas do Linux, fugindo do escopo desta monografia, que aborda sobre a ambientação do usuário apresentando os comandos e procedimentos básicos do Linux.

² Guia Foca Linux: <http://focalinux.cipsga.org.br/>

3 Metodologia Utilizada para Criação das Tarefas.

Este trabalho não se resume num amontoado de tarefas para o aprendizado do Linux jogadas de qualquer forma. Todo o processo de criação das tarefas, desde a escolha de quais comandos e procedimentos que fazem parte, a definição da seqüência dos passos das tarefas até a aplicação prática do trabalho foram cuidadosamente elaboradas.

Desta forma, na elaboração deste trabalho houve a preocupação em apresentar a metodologia e principais decisões que definiram a criação dessas tarefas. Dentre estes podem ser destacados:

- Direcionado para iniciantes em Linux;
- Escolha dos comandos e procedimentos mais comuns e utilizados no cotidiano;
- Criação das tarefas em ordem cronológica e de dificuldade;
- Aproveitamento do conhecimento já apresentado no desenvolvimento de tarefas anteriores;
- Teste e execução de cada tarefa nas distribuições Linux: SUSE, Mandrake e Debian;
- Uso do modo texto;
- Resolução de utilizar as tarefas para incentivo a pesquisa.

Nas subseções a seguir são apresentados em detalhes os principais elementos do processo de definição e criação das atividades e tarefas tratadas nesta proposta para aprendizado do Linux.

Neste trabalho, atividade e tarefa tem significados distintos. Atividade é um conjunto de uma ou mais tarefas para uma mesma finalidade, por exemplo: “manipulação de arquivos” é uma atividade que contém uma série de tarefas para apresentação dos diversos comandos para manipulação de arquivos (`rm`, `cp`, `mv`, `touch`, `cat`, `tail`, `head`, `more`, `less` e `rename`), sendo que para cada comando pode existir uma ou mais tarefas. Por tarefa entende-se cada passo de uma atividade.

3.1 Uso do Modo Texto

A proposta para aprendizado do Linux está construída tendo por base o uso do modo texto, apresentando apenas algumas pequenas indicações do modo gráfico. O uso da interface gráfica com certeza é bem mais amigável, principalmente para usuários comuns. Entretanto, não é possível usar o Linux eficazmente caso não se saiba como usar a linha de comando e utilitários em modo texto. E em se tratando de profissionais de TI, pode-se dizer que gerenciar o Linux sem conhecer o modo texto é quase impossível.

Em algumas situações, a linha de comando oferece a única maneira de interagir com o sistema. Como nos casos de acesso remoto a um servidor por meio de uma conexão *dial-up* ou de uma rede, ou até mesmo para acesso a algum servidor que não possua interface gráfica. Além do mais, as ferramentas de linha de comando têm mais opções do que as ferramentas em interface gráfica

e são mais rápidas porque não incluem uma maior exigência de processamento, como por exemplo, o que ocorre no *desktop*: GNOME ou KDE. Desta forma, pode-se dizer que para quem deseja realmente aprender sobre o sistema operacional Linux, o melhor caminho é através da linha de comando.

Além disso, enfatizar o uso do modo texto nesta proposta de aprendizado do Linux tem por objetivo desmistificar o seu uso, que é tido como uma das “grandes dificuldades” no manuseio do Linux. Também se pretende propiciar a estes usuários o aprendizado de algo mais do que a simples habilidade de apontar-e-clicar com o *mouse*.

3.2 Escolha dos Comandos

Os comandos que fazem parte deste trabalho são os mais conhecidos do Linux e utilizados para as atividades do dia-a-dia. Os comandos menos utilizados ou mais complexos não são abordados neste trabalho.

Para escolha de quais comandos fariam parte do trabalho, primeiro houve a elaboração de uma lista das atividades corriqueiras, as necessárias desde o reconhecimento do sistema, o acesso ao sistema, operações básicas com arquivos e diretórios, básico de gerenciamento do sistema, até a saída e desligamento do sistema. Esta lista foi organizada em ordem cronológica de execução natural dos procedimentos levando em conta os diferentes níveis de dificuldade. Por exemplo:

- A atividade de efetuar o *logon* no sistema está após o reconhecimento das múltiplas sessões e antes da atividade de desligar ou reiniciar o sistema. Desta forma leva em

consideração a ordem cronológica da atividade.

- As atividades de reconhecimento do sistema estão na fase de ambientação, as que tratam das atividades mais corriqueiras estão no básico e as atividades iniciais para gerenciamento do sistema estão na fase intermediária, levando em consideração o nível de dificuldade para execução.

Desta forma, com base na lista de atividades e após pesquisa em referências como (ARAUJO, 2001), (NEVES, 2005), (TIBET, 2001), (SILVA, 2005) e (VEIGA, 2004), os procedimentos e comandos necessários para a execução de cada atividade da lista foram selecionados. A lista das atividades com seus respectivos comandos e exemplos de tarefas, já na ordem em que foram desenvolvidas, são:

- **Conceitos básicos.** A primeira atividade relacionada é apenas teórica que tem por objetivo colocar o usuário a par dos conceitos básicos do *software* livre e Linux.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Pesquisar sobre o surgimento do Linux e responder quem foram Richard M. Stallman e Linus Benedict Torvalds.

- **Reconhecer as múltiplas sessões do Linux.** Aqui é a apresentação de uma característica própria dos sistemas Unix/Linux, muito diferente dos sistemas mais populares. E não tem como esta atividade não estar no topo da lista, como o primeiro item a ser visto pelo iniciante, pois é a porta de entrada do sistema, todo o restante está “dentro” das interfaces. Não existem comandos

específicos para esta atividade, apenas os procedimentos para reconhecimento das múltiplas sessões desse SO.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Após ligar a estação, observar o resultado quando pressionadas juntas as teclas abaixo:

<CTRL> + <ALT> + <F1>

<CTRL> + <ALT> + <F2>

<CTRL> + <ALT> + <F3>

<CTRL> + <ALT> + <F4>

<CTRL> + <ALT> + <F5>

<CTRL> + <ALT> + <F6>

<CTRL> + <ALT> + <F7>

Descrever em que modo (gráfico ou texto) cada sessão trabalha.

- **Fazer o primeiro *logon*** é a segunda atividade a ser executada, pois se a interface é a porta de entrada, o *logon* é a chave para abrir esta porta. Os comandos apresentados nesta atividade são: *exit* e *logout*.

Exemplo de tarefa para esta atividade: **Tarefa 5** – Fazer *logon* numa das interfaces texto, com os dados do superusuário *root* e em outra com os dados de usuário comum. Para navegar entre as sessões utilizar as teclas <CTRL> + <ALT> + <F1>; <CTRL> + <ALT> + <F2>; e assim por diante.

- **Desligar e reiniciar o sistema.** Após adquirir o conhecimento de como “entrar” no sistema o próximo passo é aprender como desligar e reiniciar o sistema. Os comandos utilizados nesta atividade são: `shutdown`, `halt`, `poweroff` e `reboot`.

Exemplo de tarefa para esta atividade: **Tarefa 3** – Pela interface texto. Testar o comando `reboot` com o usuário comum e com o superusuário `root` e descrever o resultado. Com qual dos usuários houve efetivamente a reinicialização do sistema? (da mesma forma que para desligar, em algumas distribuições será possível executar como usuário comum, mas o correto é funcionar somente para o superusuário `root`). Fazer o mesmo teste com o comando: `shutdown -r now`

- **Reconhecer o ambiente próximo.** Após entrar no sistema, haverá a necessidade de uma ambientação. Desta forma, os comandos escolhidos são os que apresentam o diretório local (comando `pwd`), mostra o que existe (arquivos e diretórios) neste local (comando `ls`) e possibilita a movimentação, sair do local que está e entrar em outros “ambientes” (diretórios) do sistema (comando `cd`).

Exemplo de tarefa para esta atividade: **Tarefa 1** – O comando `pwd` mostra o diretório atual. Executar o comando na sessão logado como superusuário `root` e na outra sessão logado como usuário comum. Observar a diferença entre os resultados.

- **Uso dos poderes do superusuário `root`.** Esta atividade está relacionada aqui porque se trata de um item de segurança importante. Antes que

o iniciante em Linux crie o péssimo hábito de ficar permanentemente logado como superusuário *root*, para facilitar a execução de uma série de atividades, é apresentada bem cedo uma opção segura de conseguir poderes do superusuário *root* sem o *logon* permanente. Nesta atividade é apresentado o comando `su`.

Exemplo de tarefa para esta atividade: **Tarefa 4** – Até agora para executar algum comando como o superusuário *root*, foi solicitado que fosse feito um *logon* com este usuário. O comando `su` facilita o uso dos poderes deste superusuário sem necessidade de abrir uma nova sessão.

- Logado como usuário comum, executar o comando `su`. Entrar com a senha do superusuário *root* e observar o que acontece.

- Testar executando algum dos comandos anteriormente apresentados que só funcionaram para o superusuário *root*. Veja também a alteração do *PROMPT*.

- Para voltar ao usuário comum executar `exit`.

- **Limpar a tela.** O comando para esta atividade é o `clear`. Este comando simples efetua a limpeza da tela, facilitando a visualização na execução de novos comandos, pois somente fica na tela o resultado do comando posterior ao `clear`. A decisão por apresentar este comando na ambientação, deve-se justamente por proporcionar esta visualização do conteúdo da tela, o que facilita para os iniciantes.

Exemplo de tarefa para esta atividade: **Tarefa 5** – Este é um comando simples, utilizado para limpar a tela. Executar o comando `clear` e observar o resultado. Este comando facilita a visualização na execução de novos comandos, pois somente fica na tela o resultado do comando posterior ao `clear`.

- **Reconhecer o ambiente todo.** Antes de iniciar uma série de comandos faz-se necessário conhecer o ambiente todo do sistema Linux. Desta forma, aqui é apresentado uma série de procedimentos que propiciam o reconhecimento da árvore de diretórios do Linux. Os comandos utilizados nesta atividade são os já conhecidos: `ls`, `pwd`, `cd` e `cat`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Para visualizar a árvore principal de diretórios, executar na seqüência os comandos:

```
cd /      (muda para o diretório raiz, que é representado pela barra "/")
ls -l
```

Tarefa 2 – Pesquisar e descrever sobre o que compõe cada diretório da árvore principal de diretórios do Linux, visualizado na tarefa anterior.

- **Reconhecer algumas particularidades do Linux.** Esta atividade complementa a ambientação, uma vez que envolve o conhecimento do uso do <TAB> para completar comandos, a existência de um histórico dos comandos executados, visualização dos resultados dos comandos que ultrapassam o limite da tela e *case sensitive* (diferencia letras maiúsculas e minúsculas). Para esta atividade os comandos são os já conhecidos: `ls`, `cd` e `pwd`.

Exemplo de tarefa para esta atividade: Tarefa sobre o uso das teclas <SHIFT> + <Page Up> e <Page Down> - **Tarefa 3** – No modo texto, executar a seqüência dos comandos abaixo e observar a utilidade das teclas <SHIFT> + <Page Up> e <SHIFT> + <Page Down>.

```
cd /home      ls -l
```

```
cd /          ls -l
cd           ls -la
```

É bem provável que após executar todos os comandos acima, o resultado do primeiro comando `ls -l` (executado para o diretório */home*) não esteja mais aparecendo na tela. Visualizar novamente o resultado deste comando sem executar o comando, apenas navegando pela tela utilizando as teclas <SHIFT> + <Page Up> e <SHIFT> + <Page Down>.

- **Obter ajuda.** Esta atividade foi colocada nesta seqüência porque após o reconhecimento do ambiente e antes de começar a utilizar o sistema é necessário saber como conseguir ajuda e como encontrar a documentação do Linux. Os comandos selecionados para estas atividades são os comandos de ajuda do Linux, ou seja: `man`, `info`, `help`, `locate`, `slocate`, `which`, `apropos` e `whatis`. Também é apresentado como encontrar a documentação dos aplicativos instalados no sistema, os *HOWTO's* e a pesquisa na Internet.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Utilizando o manual (comando `man`), descobrir como executar o comando `cd` para as solicitações abaixo: (`man cd`).

a - Se posicionar no diretório raiz. (`cd /`).

b - Voltar ao diretório *home* sem utilizar o caminho do diretório.

c - Voltar ao diretório anterior (no caso o raiz) sem usar o comando utilizado no item a. Existe um parâmetro para o comando `cd` que retorna ao diretório anterior ao atual.

d – Se posicionar no diretório `/usr/sbin`.

e – Ir ao diretório `/usr/bin` sem utilizar o caminho completo do diretório, basta no próprio comando `cd`, voltar ao diretório `usr` e depois entrar no diretório `bin`, com apenas um comando. Para voltar um diretório pode ser usado dois pontos `..`, (`cd ..`).

Obs.: Para verificar o diretório atual sempre utilizar o comando `pwd`. Observar também as explicações sobre caminho absoluto e relativo, já apresentados. Para sair do manual basta pressionar a tecla `q` e para obter mais informações sobre o `man` basta executar: `man man`.

- **Manipulação de arquivos e diretórios.** Como tudo gira em torno de arquivos e diretórios dentro de qualquer sistema operacional, não sendo diferente no Linux, aprender a manipular estes arquivos e diretórios precede qualquer outra atividade após a ambientação do sistema. Esta atividade está subdividida em: reconhecer extensões de arquivos; reconhecer arquivos ocultos; criar, remover, copiar e mover diretórios; criar, visualizar, copiar, mover, remover e renomear arquivos. Os comandos novos utilizados são: `mkdir`, `rmdir`, `rm`, `cp`, `mv`, `touch`, `cat`, `tail`, `head`, `more`, `less` e `rename`.

Exemplo de tarefa para esta atividade: **Atividade 3** – Usando o comando `mkdir`, criar um diretório chamado **teste** dentro do diretório `home`. Visualizar o diretório recém criado utilizando o comando `ls`.

- **Reconhecer e manipular permissão de acesso a arquivos e diretórios.** Esta atividade leva a um reconhecimento de como funciona a permissão no Linux. Está após a atividade relacionada a arquivos e diretórios, pois se trata de um complemento a manipulação dos mesmos. Além dos

comandos já conhecidos também são utilizados os comandos `chmod`, `chgrp` e `chown`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Utilizando o comando `cd` se posicionar no diretório `/home` do usuário logado. Executar o comando `ls -l` e observar os dados apresentados dos arquivos e diretórios. Notar que o primeiro dígito da linha é diferente podendo ser `(-)` ou `(d)`, o que indica esse dígito?

- **Criação de atalhos.** Também por estar relacionado a arquivos e diretórios está nesta seqüência. Além dos comandos já conhecidos são utilizados os comando `ln` e `ln -s`.

Exemplo de tarefa para esta atividade: **Tarefa 3** – Criação de um *symbolic link*.

- Utilizando o comando `mkdir` criar um diretório chamado **atalho1** dentro do diretório `home` do usuário. (antes se posicionar no diretório `home` utilizando o comando `cd`).

- Utilizando o comando `cp` copiar o arquivo `/etc/passwd` para dentro do diretório **atalho1**. Visualizar a cópia do arquivo com o comando `ls -l ./atalho1`.

- Utilizando o comando `ln -s` e posicionado no diretório `home` criar um *symbolic link* com nome `sl_passwd` apontando para o arquivo `passwd` copiado para o diretório **atalho1**.

- Pelo comando `ls -l` e `ls -l ./atalho1` verificar a diferença deste *link* recém criado para os demais arquivos. Observar o tamanho do *link* em relação

ao arquivo original.

- Executar `cat sl_passwd` e `cat ./atalho1/passwd` e observar o resultado.
- Utilizando o comando `ls -i` visualizar o *inode* do *link* `sl_passwd` e do arquivo `./atalho1/passwd`.
- Utilizando o comando `rm` excluir o arquivo *passwd* que está dentro do diretório **atalho1** (`rm ./atalho1/passwd`).
- Pelo comando `ls -l` observar se houve alterações no *link* `sl_passwd`.
- Executar novamente o comando `cat sl_passwd`.
- Utilizando o comando `cp` copiar novamente o arquivo *passwd* para dentro do diretório **atalho1** e observar se há alterações pelos comandos `ls -l` e `cat sl_passwd`.
- Visualizar novamente o *inode* dos dois arquivos utilizando o comando `ls -i`.

- **Edição de textos** – No sistema Linux existe uma grande quantidade de arquivos textos, sendo uma atividade importante o domínio de um editor de texto. A escolha do editor de texto Vim para esta atividade, deve-se ao fato do mesmo ser um dos mais conhecidos e presentes como padrão na grande maioria das distribuições.

Exemplo de tarefa para esta atividade: **Tarefa 3** – Observar que ao executar o editor de texto Vim, o mesmo é iniciado em modo de comando. Para começar a digitar o texto será necessário alterar para o modo de inserção pressionado no teclado a letra “i”. Observar a informação “INSERT” no

rodapé do arquivo. Após entrar em modo de inserção digite o texto abaixo:

Educação é uma coisa admirável.
Mas é sempre bom lembrar,
De tempos em tempos, que nada daquilo
Que realmente vale a pena saber
Pode ser ensinado.

Oscar Wilde

Tarefa 6 – Após digitar o texto acima, sair do arquivo salvando. Pressionar a tecla <ESC> para entrar no modo comando e digitar o comando para salvar e sair (:wq).

- **Manipulação de textos.** Esta atividade complementa a anterior. Após o aprendizado da edição de textos, nada mais natural que aprender a manipular estes textos. Os comandos apresentados para esta atividade são: `grep`, `wc`, `sort`, `cut`, `nl` e `diff`.

Exemplo de tarefa para esta atividade: **Tarefa 1** - Utilizando o comando `grep`, procurar pela ocorrência “/bin/sh” no arquivo `/etc/passwd`, visualizando o resultado da ocorrência com o número da linha.

Obs.: O comando `grep` pode ser utilizado também para filtrar saídas de outros comandos. Por exemplo: `ls /etc | grep resolv` (lista o diretório `/etc` com o comando `ls` e direciona a saída do comando `ls` para o comando `grep`, que por sua vez procura pela expressão `resolv`). Para

arquivos compactados existe o comando `zgrep`, que funciona exatamente da mesma forma que o comando `grep`.

- **Compressão e empacotamento de arquivos e diretórios.** Esta atividade foi selecionada porque nos sistemas Linux praticamente todos os pacotes são distribuídos empacotados ou comprimidos, senão ambos. O que faz necessário este conhecimento como base inicial. Os comandos utilizados nesta atividade são: `compress`, `tar`, `zip`, `gzip`, `unzip` e `gunzip`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – O comando `tar` serve tanto para empacotar como para desempacotar arquivos, com ou sem opção para compressão. Por exemplo, para empacotar todos os arquivos `.conf` do diretório `/etc`, é utilizado o comando: `tar -cvf conf.tar /etc/*.conf`. Utilizando o comando `tar`, visualizar o conteúdo do arquivo `conf.tar` e desempacotar os arquivos em outro diretório.

- **Verificar a data e a hora do sistema.** Esta atividade não poderia faltar para complementar o conhecimento básico em Linux, pois o sincronismo da data e hora do sistema com o momento atual é crucial para muitos aplicativos que utilizam a data e hora do sistema, para tarefas agendadas e arquivos de registro (*log*). Os comandos apresentados nesta atividade são: `date` e `cal`.

Exemplo de tarefa para esta atividade: **Tarefa 4** – Utilizando o comando `cal` visualizar o calendário para o ano atual. Direcionar o resultado do comando para dentro de um arquivo chamado `ano`. Visualizar o conteúdo do arquivo utilizando o comando `cat`.

- **Criação de contas e grupos de usuários** – Esta atividade dá início ao grupo de atividades mais voltadas para administração do sistema. O básico da administração do sistema inicia com o gerenciamento dos usuários, portanto esta primeira atividade leva ao conhecimento básico para criação de contas e grupos de usuários na estação local. Os comandos utilizados são: `useradd` ou `adduser`, `passwd` e `addgroup` ou `groupadd`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Utilizando o comando `adduser` ou `useradd`, cadastrar um novo usuário no sistema utilizando os padrões definidos no sistema.

Tarefa 2 – Utilizando o comando `passwd` atribuir uma senha para o usuário criado na tarefa anterior.

- **Exclusão de contas e grupos de usuários** – Esta atividade é o complemento da anterior, uma vez que os usuários e grupos são criados, existe a necessidade de saber como eles são excluídos do sistema. Os comandos novos para esta atividade são: `userdel` e `groupdel`.

Exemplo de tarefa para esta atividade: **Tarefa 7** – Utilizando o comando `userdel`, excluir o usuário criado na tarefa 6. Executar `cat /etc/passwd` e verificar se o usuário foi excluído do arquivo. Verificar no diretório `/home` se o diretório `home` do usuário foi excluído.

- **Administração de contas e grupos** – Esta atividade está nesta seqüência porque complementa as duas anteriores, uma vez que usuários e grupos existem no sistema, terão que ser administrados. Nesta atividade são

apresentados os comandos: `id`, `usermod`, `logname`, `users`, `groups`, `last`, `w`, `finger`, `who` e `whoami`.

Exemplo de tarefa para esta atividade: **Tarefa 12** – Utilizando o comando `addgroup` ou `groupadd` criar um novo grupo denominado **contabilidade**. Visualizar a criação do grupo pelo comando `cat /etc/group`.

Tarefa 13 – Utilizando o comando `usermod` atribuir ao novo usuário o grupo **contabilidade** criado na tarefa anterior. Visualizar quais foram as alterações pelos comandos `cat /etc/passwd` e `cat /etc/group`.

- **Gerenciamento de processos** – Como no Linux, qualquer programa em execução é considerado um processo, torna-se necessário para qualquer usuário ter o conhecimento básico para tratar com estes processos. Esta atividade apresenta diversas formas para visualizar os processos em execução no sistema e como “matar” estes processos. Esta atividade está nesta ordem porque dentre as atividades de administração é uma das mais úteis para o dia-a-dia de qualquer usuário, portanto é considerada uma atividade básica para administração do sistema. Os comandos novos apresentados são: `ps`, `pstree`, `top`, `kill` e `killall`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Utilizando o comando `ps` listar de forma resumida, todos os processos do usuário logado.

- **Gerenciamento do sistema** – Esta atividade apresenta alguns comandos úteis para verificação da situação do sistema como memória, espaço

em disco, espaço utilizado por determinado arquivo/diretório, quanto tempo o sistema está ativo e o acesso a outros dispositivos como disquete e CD. Está nesta seqüência porque contém uma série de comandos básicos e importantes para a utilização e verificação do sistema, complementa a tarefa anterior na administração do sistema. Os comandos utilizados são: `free`, `df`, `du`, `uptime`, `mount` e `umount`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Utilizando o comando `free`, verifique a quantidade de memória livre e utilizada pelo sistema. Execute novamente exibindo a quantidade em *Mbytes* e com totalizador.

Tarefa 2 – O comando `df` exibe a quantidade de espaço utilizado pelos sistemas de arquivos. Utilizando o comando `df` verificar qual espaço utilizado pelos sistemas de arquivos atualmente montados no sistema.

- **Criando apelidos para comandos (Alias)** – Esta atividade está nesta seqüência porque complementa a administração do sistema e é tratada em separado porque dá início a administração pela configuração, ou seja, para criação de apelidos é necessário efetuar alteração nos arquivos do sistema. O comando apresentado nesta atividade é o `alias`.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Efetuar *logon* como usuário comum. Utilizando o comando `alias` criar o apelido `cls` para o comando `clear` (`alias cls='clear'`). Testar a execução do `clear` utilizando o apelido `cls`. Digitar no *PROMPT* apenas `cls`.

Tarefa 2 – Efetuar *logon* com outro usuário em outra interface texto e testar a execução do `cls`. Houve a execução do comando como na tarefa 1?

Tarefa 3 – Utilizando o comando `reboot` reiniciar a estação. Após efetuar *logon* com o mesmo usuário da tarefa 1. Testar novamente a execução do `cls`. Houve a execução do comando como na tarefa 1?

- **Agendamento de tarefas** – Esta atividade foi escolhida por tornar o dia-a-dia mais fácil para execução de processo com hora marcada, sendo, portanto básico para administração do sistema, e foi colocada nesta seqüência por ser uma atividade um pouco mais complexa que as anteriormente apresentadas. Os comandos para esta atividade são: `at` e `crontab`.

Exemplo de tarefa para esta atividade: **Tarefa 2** – Utilizando o `crontab`, agendar a reinicialização do sistema para que ocorra a 05 minutos do momento atual. Acompanhar a execução do agendamento. Obs.: Fazer o agendamento diretamente no arquivo `/etc/crontab` e logado como usuário *root*.

- **Shell Script** – Esta atividade foi incluída neste trabalho apresentando apenas o básico do básico para construção de *Shell Script*. O desenvolvimento de *Shell Script* não é considerado básico, mas o entendimento do mesmo o é. Nesta atividade somente são apresentados os procedimentos para criação e execução de um *Shell Script* bem simples.

Exemplo de tarefa para esta atividade: **Tarefa 2** – A primeira informação de um *Shell Script* é qual interpretador será utilizado, após esta informação todos os comandos serão executados segundo aquele interpretador informado exceto as linhas que começarem com `"#"`. Para entender de forma fácil como isto

funciona, os passos abaixo orientam a criação e execução de um pequeno *Shell Script*.

- Se posicionar no diretório *home* utilizando o comando `cd`.

- Utilizando o editor de texto `Vim`, criar um arquivo com nome de `ola.sh` com o conteúdo abaixo:

```
#!/bin/bash

# Meu primeiro Shell Script

clear

echo ""

echo ""

echo "   Ola Mundo !!!!! "
```

- Salvar e sair do arquivo (**`:wq!`**).

- Para execução de arquivos no Linux é necessário dar permissão para execução. Utilizando o comando `chmod`, dar ao arquivo `ola.sh` a permissão para execução ao dono. (`chmod 744 ola.sh`).

- Utilizando o comando `ls`, verificar as permissões do arquivo `ola.sh`.

- Executar o *Shell Script*. Observar que se apenas informar na linha de comando o nome do arquivo "`ola.sh`", o Linux retornará a informação de que não encontrou o comando. Neste caso é necessário informar o caminho do diretório em que o arquivo está para que o Linux encontre o arquivo. Executar, portanto desta forma: `./ola.sh`.

- **Equivalências do *Windows/DOS* e *Linux*** – Esta atividade foi incluída neste trabalho para facilitar a vida dos usuários que estão migrando do sistema *Windows* para o *Linux*. São apresentadas apenas algumas equivalências entre os comandos do *DOS* e do *Linux* e induz a pesquisa por equivalentes para alguns dos aplicativos mais utilizados.

Exemplo de tarefa para esta atividade: **Tarefa 1** – Pesquisar quais são os equivalentes no *Linux* para os aplicativos relacionados abaixo:

- *Office* –
- *Internet Explorer* –
- *Outlook* –
- *SQL* –

Como pode ser observado, os comandos escolhidos para este trabalho são os mais conhecidos e utilizados para as atividades corriqueiras, pois o objetivo do trabalho é propiciar uma ambientação a novos usuários do *Linux*. Após contato com estes comandos básicos espera-se que o aprendiz à medida que suas necessidades vão surgindo ampliem o leque de conhecimento desses comandos mediante pesquisa em material didático.

Um exemplo de escolha de comando: (VEIGA, 2004) apresenta vinte e oito utilitários de texto (**cat**, *colrm*, *column*, **cut**, *expand*, *fmt*, *fold*, **grep**, *egrep*, *fgrep*, **head**, *hexdump*, *join*, **less**, *look*, **more**, **nl**, *paste*, *ver*, **sort**, *tac*, *tail*, *tee*, *ul*, *unexpand*, *uniq*, *zmore* e *zgrep*). Mas nesta proposta apenas são utilizados dez utilitários (*grep*, **wc**, *sort*, *cut*, *nl*, *head*, **less**, *more*, *tail* e **diff**) que estão diluídos

entre as tarefas para manipulação de arquivos e diretórios e as tarefas para utilitários de texto. Destes dez utilitários tratados nesta proposta, dois são considerados por (VEIGA, 2004) como comandos para arquivos e diretórios, portanto da lista de vinte e oito utilitários, apenas oito são tratados neste trabalho destacando somente os comandos mais conhecidos e úteis para uma ambientação no sistema Linux, os quais fazem parte desta seleção.

3.3 Criação das Tarefas

A criação de cada tarefa desta proposta para aprendizado do Linux tem como ponto de partida três princípios básicos: O ponto de vista do iniciante em Linux; a efetiva execução da tarefa no Linux e o incentivo pela busca por informações (pesquisa).

- No que diz respeito ao **ponto de vista do iniciante em Linux** – A criação de cada tarefa segue a mesma filosofia da criação das atividades, isto é, para criar a relação das atividades houve a simulação de um iniciante em Linux acessando o sistema pela primeira vez. De igual forma, na criação de cada tarefa que compõe a atividade ocorre a simulação da execução de cada comando ou procedimento. Desta forma são privilegiados neste trabalho os comandos e procedimentos básicos que seriam necessários para que o iniciante acesse o sistema e conheça como executar os comandos e procedimentos básicos no Linux.

- Para a **efetiva execução das atividades** – as tarefas foram construídas de forma que o aprendizado ocorra pela execução dos comandos diretamente no sistema operacional Linux.

A decisão por traçar um projeto de aprendizado que utiliza tarefas para aplicação prática deve-se principalmente ao fato de que existe uma melhor fixação do conteúdo quando o aprendiz vivencia a execução dos comandos diretamente no Linux apreciando seus resultados.

A este respeito cabe ser destacado que um dos princípios da reconstrução cognitiva que baseia o construtivismo afirma que a aprendizagem é alcançada somente quando uma habilidade ou conceito é cristalizado. Respeitando esses princípios essa proposta de aprendizagem precisa incluir atividades práticas significativas que estimula o processo ensino-aprendizagem humano.

Esta proposta de aprendizado do Linux é baseada essencialmente na prática para que o conhecimento seja cristalizado. Reforça aqui a teoria do construtivismo que diz que o indivíduo tem papel ativo na construção do conhecimento, conforme o autor construtivista Lev Semenovitch Vygotsky como comentado em (UCHÔA, 2001) e (BEN-HUR, 2005).

- O princípio básico que **incentiva a busca por informações (pesquisa)** foi considerado na elaboração da tarefa, sendo ela construída de forma a não conter todas as informações necessárias para sua execução, o que leva o aprendiz a fazer pesquisa em material complementar. A fim de facilitar a pesquisa, nas próprias tarefas existem indicações de fontes para pesquisa, conforme pode ser verificado no Apêndice C.

A decisão em construir a proposta desta forma leva em consideração os benefícios para qualquer profissional que tenha desenvolvido em seu local de trabalho um espírito de pesquisa. O espírito de pesquisa enriquece o trabalho, que poderá contar com melhores práticas para resolução de qualquer desafio que seja proposto. Desta forma, mesmo sendo esta proposta de aprendizado de um sistema operacional, direcionado ao perfil de profissionais de TI, não somente

estes podem se beneficiar com esta indução a pesquisa.

Mesmo existindo uma indicação de bibliografia junto às tarefas, o aprendiz tem a total liberdade para pesquisar no material que melhor convier. Esta liberdade também tem a ver com os problemas enfrentados no dia-a-dia, quando nenhuma mensagem de erro traz junto a indicação de como conseguir a solução para o problema que a ocasionou. Desta forma, esta proposta induz o aprendiz a buscar pela resolução dos problemas que são apresentados.

3.3.1 Processo de Criação das Tarefas

O processo para a criação das tarefas partiu da listas de atividades pronta, a mesma lista utilizada para definição dos comandos, já com a indicação dos comandos e procedimentos a serem tratados. Para cada seqüência proposta nesta lista de atividades, houve a preocupação em seguir este procedimento: Pesquisa sobre o comando em questão; criação da seqüência dos passos para aquele comando e efetivo teste da tarefa no Linux (SUSE, Mandrake e Debian).

- A **pesquisa sobre o comando em questão** foi realizada com base nas referências (ARAUJO, 2001), (NEVES, 2005), (TIBET, 2001), (STATO, 2004), (BALL, 2004), (MARQUES, 2004) (JANG, 2003), (SILVA, 2005) e (VEIGA, 2004).

Além de verificar o que cada autor apresenta sobre o comando em questão, o mesmo foi testado com diversos de seus parâmetros diretamente no

Linux, para observar o comportamento do mesmo e definir o que é básico e essencial para apresentar na tarefa sobre aquele determinado comando.

A passagem para o próximo passo do processo de criação das tarefas ocorre somente após a coleta e estudo das informações sobre o comando e a efetiva vivência da execução do mesmo no sistema operacional.

Por exemplo: Para criação da tarefa sobre desligar e reiniciar o sistema, os comandos `shutdown`, `halt`, `poweroff` e `reboot`, foram estudados nas referências apresentadas e efetivamente executados no Linux com todos os parâmetros possíveis e com usuários diferentes. Após este procedimento foi selecionado o que é básico e necessário para um iniciante conhecer nesta fase do aprendizado. Somente após este estudo, teste e avaliação do comando o processo de criação das tarefas relativas àquela atividade passou para a fase seguinte.

- **Criação da seqüência de passos** – A criação da seqüência da tarefa inicia com a simulação da execução do comando no Linux. Este processo leva em consideração o que é necessário para um iniciante em Linux e quais informações já foram repassadas em tarefas anteriores. Fruto da fase anterior, em que o comando foi estudado e testado no Linux.

Com estas informações e simulando a execução do comando ocorre o desenvolvimento de uma seqüência de passos em ordem cronológica de execução para aquele determinado comando. Esta seqüência de passos se transforma na tarefa proposta.

Depois da criação da seqüência dos passos para aquela determinada tarefa os mesmos são ao mesmo tempo validados (executados) no Linux, enquanto a tarefa é desenvolvida. Assim, a tarefa não propõe algo que

efetivamente não será possível de executar uma vez que a proposta é feita a partir de uma execução do comando diretamente no Linux.

Por exemplo: Para a criação da primeira tarefa prática (reconhecendo as múltiplas sessões do Linux), ocorre a simulação de um iniciante em frente a uma estação com Linux pela primeira vez. A partir desta simulação, há a percepção da necessidade de que a primeira informação sobre Linux seja a possibilidade de se trabalhar com múltiplas sessões. O primeiro passo foi a pesquisa na bibliografia já apresentada sobre como cada autor aborda o assunto. Na seqüência os procedimentos são executados no Linux e a partir desta execução ocorre a definição da seqüência de passos necessários para que um iniciante execute para reconhecer as múltiplas sessões do Linux. Com os passos definidos as tarefas necessárias são desenvolvidas e testadas para validação.

Tendo por base esses procedimentos, todas as tarefas foram criadas. Aproveitando como base a lista de atividades já com os comandos definidos. Em seguida parte para a pesquisa sobre os comandos, depois ocorre a simulação de uso e criação/definição da seqüência dos passos necessários para visualização do potencial de cada comando ou procedimentos relacionado, sempre levando em consideração o ponto de vista do iniciante em Linux.

Para auxílio em muitas atividades houve o acréscimo de algumas dicas que aparecem quando há um aumento no grau de dificuldade ou orientar o iniciante para melhor utilização dos recursos disponíveis.

Exemplo de dica para melhor utilização dos recursos: Dica apresentada para o comando `man`: Abrir duas sessões em modo texto, numa sessão executar o comando `man` e na outra testar os comandos. Conforme pode ser verificado na tarefa em questão que encontra-se no Apêndice C.

Exemplo de dica quando o grau de dificuldade é maior: Dica apresentada para o agendamento de *Shell Script*: Além do arquivo do *Shell Script* ter as permissões corretas para executar, será necessário informar o caminho completo do diretório em que o *script* está armazenado. Conforme pode ser verificado na tarefa em questão que encontra-se no Apêndice C.

- **Teste da tarefa no Linux** – O teste das tarefas no Linux serve para validar a efetiva possibilidade de execução de cada tarefa. Além das tarefas serem criadas diretamente no Linux, as mesmas foram testadas em outras distribuições. As distribuições escolhidas para teste das atividades foram: Mandrake, SUSE e Debian. Mesmo assim, o trabalho pode ser utilizado em outras distribuições, uma vez que foram utilizados apenas os comandos básicos. A distribuição em que as tarefas foram criadas foi a Mandrake, somente depois as tarefas criadas foram testadas nas demais distribuições.

O teste das tarefas consiste basicamente em seguir os mesmos passos que seguiria um iniciante em Linux, executando a seqüência de passos proposta nas atividades. Após a definição de quais atividades seriam tratadas na proposta, escolha dos comandos para cada atividade, como seria o desenvolvimento das tarefas, definido a seqüência de passos para elas, ocorre a definição da ordem em que estas atividades e tarefas serão apresentadas.

Para o arranjo das atividades com suas respectivas tarefas foram observadas duas variáveis importantes, a ordem cronológica da atividade e o grau de dificuldade das mesmas. Pode ser observada em cada atividade a presença da ordem cronológica ou do grau de dificuldade ou ambas. Por ordem cronológica entende-se a ordem natural de execução das atividades, por exemplo: Primeiro é necessário efetuar o *logon* no sistema operacional para

depois conseguir executar qualquer comando.

Quando o grau de dificuldade de duas atividades é o mesmo, prevalece a ordem cronológica. Quando duas atividades podem ser executadas na mesma seqüência, prevalece o grau de dificuldade para execução.

Cada tarefa foi construída com a preocupação em acrescentar o novo comando e/ou procedimento definido para a tarefa em questão e reforçar na medida do possível os já apresentados em tarefas anteriores. Exemplo de uma tarefa: Um dos itens da tarefa para criação de um *hard link*: “Utilizando o comando `mkdir` criar um diretório chamado **atalho** dentro do diretório `home` do usuário. (antes se posicionar no diretório `home` utilizando o comando `cd`)”. Esta tarefa usada como exemplo está apresentando os passos para o aprendizado do comando `ln` (criação de *links*), como pode ser visto no Apêndice C. Sendo que os comandos `mkdir` e `cd`, utilizados nesta tarefa, já são apresentados em tarefas anteriores.

Desta forma existe um reforço na utilização destes comandos (`mkdir` e `cd`), o que comprova a preocupação do trabalho na recontextualização, estando, portanto em acordo com os princípios do aprendizado através da reconstrução cognitiva. O construtivismo de Piaget defende que o conhecimento é fruto de uma elaboração pessoal baseada em experiências anteriores, como apresentado em (BEN-HUR, 2005). De certa forma, o cuidado em reaproveitar em cada atividade o conteúdo já apresentado em atividades anteriores, privilegia o iniciante neste reaproveitamento de suas experiências para execução da atividade atual.

3.4 Divisão das Fases

Conforme já mencionado o direcionamento deste trabalho é para iniciantes em Linux e está dividido em três fases distintas: ambientação, básico e intermediário. O conteúdo das três fases privilegia os procedimentos corriqueiros de um usuário do Linux.

A divisão em fases separa o conteúdo em três grupos, sendo:

- A primeira fase, ambientação, apresenta os procedimentos que propiciam o **primeiro contato com o sistema operacional** e o conhecimento das características básicas e peculiares do Linux. Este grupo de atividades apresenta o sistema operacional, através de pesquisa dos conceitos básicos sobre Linux, o reconhecimento das múltiplas sessões e das características do Linux, as operações iniciais como *logon/logoff*, desligar e reiniciar o sistema e os primeiros comandos (*ls*, *cd* e *pwd*);

- A segunda fase, básico, apresenta os procedimentos para as **atividades corriqueiras**. Como obter ajuda, manipulação de arquivos e diretórios, permissão de acesso a arquivos e diretórios, criação de atalhos, editores de texto, utilitários de texto, compactação e descompactação de arquivos e diretórios, data e hora;

- A terceira fase, intermediário, apresenta os **procedimentos iniciais para administração do sistema**. Esta fase trata de assuntos como administração de usuários e grupos, gerenciamento dos processos do sistema, gerenciamento básico do sistema, configuração de alguns itens do sistema, como: Criação de atalhos (*links*); agendamento de tarefas e criação de *Shell Script*. Ainda apresenta uma relação de equivalência entre *Windows/DOS* e Linux.

A divisão destas três fases foi feita após o término da relação das atividades necessárias para que ocorresse o conhecimento das operações básicas do Linux. Com a relação pronta houve a observação que as atividades poderiam ser divididas conforme o foco principal de cada uma, o que resultou na divisão em ambientação, básico e intermediário, apresentadas nesta monografia.

4 Aplicação e Resultados

A proposta para aprendizado do Linux foi desenvolvida especialmente para um grupo específico, que faria sua utilização em grupo de estudo no laboratório da empresa.

Até o presente momento o referido laboratório ainda não está em funcionamento, portanto não foi possível validar a utilização das tarefas no ambiente e com o grupo para qual foram criadas.

Desta forma, com a finalidade de obter certa validação das atividades dessa proposta de aprendizado do Linux foi recrutado um pequeno grupo de voluntários que não tinham conhecimento do Linux. Fazem parte deste grupo: Isaac Luiz Sarah Sidou (profissional de TI), Reinaldo Antonio Valério (profissional de TI), Rubens Cruz Praude (estudante) e Marcelo Cruz Praude (estudante).

Para coleta das informações deste grupo de voluntários após a utilização da proposta para aprendizado do Linux foram utilizados dois questionários, um sobre o perfil e o outro é uma avaliação da utilização das atividades.

Dentre as formas para coleta de informações (entrevista, questionário, formulário e observação) apontadas por (BELO, 2000), o uso do questionário torna-se mais adequado, devido às distâncias entre pesquisador e pesquisados o que torna inviável a utilização dos demais métodos.

As questões foram desenvolvidas de forma livre e aberta, onde o pesquisado expressará a sua opinião sem nenhuma indução através de questões com múltipla escolha.

Normalmente questionários para avaliação não incluem informações sobre os dados pessoais dos pesquisados, mas como este questionário foi criado apenas para este pequeno grupo, este item foi acrescentado.

Os questionários podem ser encontrados nos Apêndices A e B e os questionários respondidos pelos voluntários estão no Apêndice D.

As conclusões após a análise dos questionários respondidos pelos voluntários, foram:

A proposta pode ser utilizada por pessoas sem perfil técnico, como foi o caso do Rubens e do Marcelo que não atuam na área de TI e, no entanto se beneficiaram com o uso da proposta.

O trabalho merece uma melhoria na apresentação dos comandos iniciais, o que foi indicado por Isaac que teve maior dificuldade na execução deste grupo de tarefas. O que facilitará também a ambientação inicial no modo texto, que foi indicado por Marcelo e Rubens que tiveram dificuldades com a interface texto.

Há a percepção da necessidade da inclusão da interface gráfica como aprendizado básico pela dificuldade que usuários sem perfil técnico têm em utilizar a interface texto.

A utilização das tarefas para aprendizado do Linux realmente proporcionaram um aprendizado do básico do Linux, indicado por todos os voluntários, apesar das dificuldades apontadas por eles em trabalhar na interface texto, comandos iniciais e para entendimento da árvore de diretórios.

Houve concordância de todos os voluntários quanto a existência da ordem crescente de dificuldade nas tarefas propostas.

A proposta pode ser utilizada individualmente, por autodidatas sem problemas, uma vez que todos os voluntários estudaram desta forma sem auxílio de algum orientador ou em grupo com outro voluntário.

As tarefas podem ser executadas sem problema em outras distribuições como o Kurumim e Fedora que não foram indicadas na monografia, mas utilizada pelos voluntários.

Não foi possível determinar um tempo médio para execução de todas as tarefas, pois além do grupo ser pequeno, dos quatro voluntários dois deles não finalizou o estudo.

As conclusões da análise dos questionários validam esta proposta para utilização de atividades práticas para aprendizado do Linux, que propõe:

- Utilização do trabalho individualmente (o que foi feito pelos quatro voluntários);
- Ordem crescente de dificuldade;
- Aprendizado do básico do Linux (ambientação);
- Fixação do conteúdo pela execução prática;
- Utilização em outras distribuições além das indicadas;
- Utilização de material complementar além do indicado no trabalho para pesquisa.

Da mesma forma, a análise dos questionários indica a existência de algumas lacunas que devem ser preenchidas como a melhoria na apresentação dos comandos iniciais e complementação de algumas tarefas para melhor entendimento do assunto tratado, como as que apresentam a estrutura de

diretórios.

Uma análise mais acirrada da utilização da proposta para aprendizado do Linux torna-se impossível diante de uma amostra pequena como esta, com apenas quatro pessoas. A aplicação desta proposta no laboratório da empresa e utilizando-se da observação direta da aplicação prática das tarefas pelos membros da equipe, com certeza indicará de forma mais segura os pontos fortes e os pontos fracos desta proposta.

5 Conclusão

Parte do sucesso num processo de migração está diretamente relacionado com a qualidade do treinamento dos usuários. Como os seres humanos são criaturas que se habitam, pode ser difícil fazer a transição de uma força de trabalho para um novo ambiente. Entra aí a perspicácia dos administradores em ambientar os usuários, bem antes que a migração efetivamente ocorra. De forma que não haja risco da descontinuidade desse processo dentro da empresa.

A proposta apresentada nesta monografia visa facilitar a ambientação de novos usuários no sistema operacional Linux. Mesmo tendo sido originalmente desenvolvida para um grupo específico de profissionais de TI, como ela privilegia a ambientação no Linux, pode ser útil para qualquer perfil de usuário iniciante em Linux.

A contribuição inicial deste trabalho será feita para o grupo para a qual foi desenvolvida, tornando possível a ambientação no sistema operacional Linux antes que ocorra a migração. Desta forma, proporcionando uma base sólida de conhecimento neste novo sistema, preparando o grupo para estarem aptos a galgarem maiores conhecimentos, garantindo assim a continuidade do suporte técnico hora prestado pela equipe. Além disso, apresentar a metodologia dessa proposta é importante, pois possibilita o desenvolvimento de trabalhos paralelos ou mesmo a atualização das tarefas com a evolução do Linux.

Com pouca ou nenhuma adequação de igual forma este trabalho poderá auxiliar outros grupos na ambientação do Linux.

Pela validação feita com o pequeno grupo de voluntários, descritas no Capítulo 4, provou-se que a utilização das atividades práticas realmente proporciona um aprendizado dos conhecimentos básicos do Linux. Apesar do grupo ser pequeno e alguns ainda não terem finalizado todas as atividades.

Em trabalhos futuros, pretende-se desenvolver as fases “experiente” e “avançado” que está previsto no trabalho inicial mas não estão contempladas nesta monografia.

Além da complementação das fases, pretende-se expandir as tarefas para as atividades das fases já apresentadas (ambientação, básico e intermediário) de formar a tornar as atividades mais completas, com criação de mais passos para os comandos já tratados a fim de melhorar a fixação do aprendizado.

Existe também a intenção de incluir nestas fases atividades para a interface gráfica, para possibilitar uma melhor ambientação dos usuários de setores administrativo. Para este perfil de usuário é interessante o conhecimento da interface texto, privilegiada neste trabalho, mas como o trabalho diário destes usuários é diretamente na interface gráfica, faz-se necessário uma apresentação desta interface.

Este trabalho aborda apenas o básico para ambientação de um usuário no fantástico mundo do Linux. Com essa base sólida espera-se que os aprendizes consigam alçar vôos maiores, rumo aos inúmeros recursos que este sistema operacional oferece.

6 Referências Bibliográficas

ARAÚJO, Jário. **Comandos do Linux**. Rio de Janeiro: Ciência Moderna, 2001. 155p. (Sistema operacional para microcomputadores).

BALL, Bill; DUFF, Hoyt. Primeiros Passos com o Linux. In:_____. **Dominando Linux: Red Hat e Fedora**. Tradução de João Eduardo Nóbrega. Revisão Técnica Rafael Rigues. 1. ed. São Paulo: Pearson Macron Books, 2004. cap. 5, p.88-118. (Sistema operacional de computador).

BELO, José Luiz de Paiva. INTRODUÇÃO À METODOLOGIA CIENTÍFICA. UNESP - Assis, 03/04/2000 Disponível em: <<http://www.assis.unesp.br/~egalhard/metciem.htm>> - Acesso em: 22/11/2005.

BEN-HUR, Meir. **Aprendizagem e Transferência**. Instituto Pieron. Disponível em: <<http://www.pieron.com.br/bib04.htm>> - Acesso em: 15/10/2005.

GOVERNO FEDERAL. Comitê Executivo de Governo Eletrônico. **e-PING: Padrões de Interoperabilidade de Governo Eletrônico**. Brasília, 13/07/2005. Versão 1.0. Disponível em: <<http://www.governoeletronico.gov.br/governoeletronico/>> - Acesso em: 12/10/2005.

InfoWester. **Artigo - Editor vi**. Publicado em 30/06/2003. Disponível em: <<http://www.infowester.com/linuxvi.php>> - Acesso em: 23/09/2005.

JANG, Michael. Como iniciar na linha de comando. In:_____. **Red Hat Linux 9**. Rio de Janeiro: Ciência Moderna, 2003. cap. 6, p.217-236. (Sistema operacional para computadores).

MARQUES SOBRINHO, Hélio; MARQUES, Flávia de Castro Mendes;

BASTOS, Frederico Machado; SOARES, Breno Silveira. Utilizando o Linux. In:_____. **Sistema Operacional SuSE Linux**. Belo Horizonte: Starix Informática, 2004. cap.4, p.37-148. (Sistema operacional de computador).

NEVES, Julio César. **Programação Shell Linux**. 5. ed. Rio de Janeiro: Brasport, 2005.408p. (Sistema operacional de computador).

PRESIDENCIA DA REPÚBLICA. **Planejamento Estratégico para Implementação de Software Livre**. Brasília. Disponível em: <<http://www.softwarelivre.gov.br/diretrizes/>> - Acesso em: 12/10/2005.

SASTRE, G. **Por que o Construtivismo**. Publicado em 04/08/2002. Disponível em: <<http://www.centrorefeducacional.pro.br/pqconstru.htm>> - Acesso em: 15/10/2005.

SHAH, Steve; GRAHAM, Steven. A linha de comando. In:_____. **Linux Administração – Teoria e Prática**. Rio de Janeiro: Alta Books, 2003. cap. 6, p.67-87. (Sistema operacional de computador).

SILVA, Gleydson Mazioli. **Guia Foca Linux**. Publicado em 30/07/2005. Disponível em: <<http://focalinux.cipsga.org.br/>> - Acesso em: 18/09/2005.

SLTI/ITI, CISL, CTSLL. Ministério do Planejamento, Orçamento e Gestão. Comitê Executivo do Governo Eletrônico. **Guia Livre - Referência de Migração para Software livre**. Brasília, 22/06/2004. Versão beta 0.5. Disponível em: <<http://www.governoeletronico.gov.br/governoeletronico/>> Acesso em: 12/10/2005.

STALLMAN, Richard. **O Projeto GNU**. Publicado pelo CIPSGA em: 06/2000. Tradução de Alexandre J. Thomé. Revisão de Eliane M. de Azevedo. Disponível em: <<http://www.cipsga.org.br/sections.php?op=viewarticle&artid=61>> -

Acesso em: 04/11/2005.

STALLMAN, Richard. **O Manifesto GNU**. Free Software Foundation, 21/01/2001. Tradução de Fernando Lozano. Disponível em: <<http://www.gnu.org/gnu/manifesto.pt.html>> - Acesso em: 04/11/2005.

STATO FILHO, André. Modo Texto. In:_____. **Domínio Linux: do Básico aos Servidores**. 2. ed. Florianópolis: Visual Books, 2004. cap. 7, p.71-136. (Sistemas operacionais).

TIBET, Chuck V. **LINUX Administração e Suporte**. São Paulo: Novatec, 2001. 379p. (Sistema operacional de computador).

UCHÔA, Kátia Cilene Amaral. **Aprendizagem e Informática: Uma Abordagem Construtivista**. Lavras, 2001. Disponível em: <<http://arl.ginux.ufla.br/~kacilene/educacao/index.html>> - Acesso em: 15/10/2005.

VEIGA, Roberto G. A. **Comandos do Linux: Guia de Consulta Rápida**. São Paulo: Novatec, 2004. 144p. (Sistema operacional de computador).

7 Apêndice A – Questionário perfil

Questionário para levantamento do perfil dos candidatos a utilização das atividades para aprendizado do Linux.

Questionário para levantamento do perfil

Perfil do aprendiz:

Nome: _____

Idade: _____ Sexo: _____

- 1 – Qual a sua área de atuação?
- 2 – Possui algum conhecimento do Sistema Operacional Linux? Se sim, qual?
- 3 – Possui algum conhecimento de algum outro Sistema Operacional? Se sim qual Sistema Operacional?
- 4 – Qual é a sua formação? Qual é o seu nível de escolaridade?
- 5 – O que motivou o estudo do Sistema Operacional Linux?
- 6 - Qual será a aplicação prática no seu dia a dia do Sistema Operacional Linux?

8 Apêndice B – Questionário avaliação.

Questionário de avaliação do aprendizado do Linux utilizando as atividades propostas.

Avaliação do aprendizado utilizando as tarefas

- 1 – Relacione abaixo as referências utilizadas como complemento as atividades para o aprendizado do Linux.
- 2 – Qual distribuição foi utilizada?
- 3 – Qual foi a maior dificuldade no aprendizado do Linux?
- 4 – O aprendizado foi individual (autodidata) ou foi feito em grupo (como complemento de curso ou em grupo de estudo)?
- 5 – Quantas horas semanais foram dedicadas ao estudo do Linux?
- 6 - Quanto tempo foi necessário para completar todas as atividades propostas?
- 7 – Em sua opinião, o direcionamento proposto traz um nível de dificuldade crescente no decorrer da apresentação das atividades ou não?
- 8 - Qual sua avaliação da validade da utilização das atividades para o aprendizado do Linux? (O aprendizado seria o mesmo sem a utilização das atividades?).
- 9 – Qual é a sua avaliação do seu conhecimento em Linux?
- 10 – Acrescente as observações que achar necessário sobre o uso das atividades para direcionamento do aprendizado do Linux.

9 Apêndice C – Tarefas

Neste Apêndice está sendo apresentado as tarefas propostas divididas nas três fases: Ambientação; básico e intermediário.

9.1 - Primeira Fase – Ambientação

Esta fase do aprendizado apresentação os primeiros contados com o sistema Linux. Leva a uma pesquisa dos conceitos básicos, o reconhecimento das múltiplas sessões e efetuar as operações iniciais e essenciais como *logon/logoff*, desligar e reiniciar o sistema e os primeiros comandos.

9.1.1 Conceitos básicos

Esta atividade tem por objetivo colocar o usuário a par dos conceitos básicos do Linux e *software* livre. As respostas devem ser buscadas através de pesquisa na Internet ou livros como (ARAUJO, 2001) e (SILVA, 2005).

Tarefa 1 – Pesquisar sobre o surgimento do Linux e responder quem foram Richard M. Stallman e Linus Benedict Torvalds.

Tarefa 2 – Descrever as premissas para que um *software* seja considerado “*software* livre”.

Tarefa 3 – Relacionar pelo menos cinco distribuições do Linux, citando a nacionalidade.

Tarefa 4 – Descrever a diferença entre *copyright* e *copyleft*.

Tarefa 5 – Descrever o que é e qual a importância da licença GPL e quando se aplica.

Tarefa 6 – Descrever o que é uma distribuição Linux.

Tarefa 7 – Descrever pelo menos duas vantagens e duas desvantagens do *software* livre.

Tarefa 8 – Citar duas empresas que utilizam *software* livre (públicas e/ou privadas).

9.1.2 Efetuando *logon/logoff*

9.1.2.1 Reconhecer as múltiplas sessões do Linux

Diferente dos outros sistemas os SO Unix/Linux trabalham com múltiplas sessões. Nesta atividade, será visto na prática como identificar e acessar estas sessões. Uma boa referência é (SILVA, 2005). Também podem ser encontradas informações em (BALL, 2004) e (TIBET, 2001).

Tarefa 1 – Após ligar a estação, observar o resultado quando pressionadas juntas as teclas abaixo:

<CTRL> + <ALT> + <F1>

<CTRL> + <ALT> + <F2>

<CTRL> + <ALT> + <F3>

<CTRL> + <ALT> + <F4>

<CTRL> + <ALT> + <F5>

<CTRL> + <ALT> + <F6>

<CTRL> + <ALT> + <F7>

Descrever em que modo (gráfico ou texto) cada sessão trabalha.

9.1.2.2 Interface Gráfica

O resultado da tarefa anterior mostra que existe apenas uma interface gráfica. <CTRL> + <ALT> + <F7>.

Tarefa 2 – No Linux existe a possibilidade de instalação de diversos tipos de interfaces gráficas. Pesquisar quais são e quais as mais utilizadas.

Tarefa 3 – Efetue o *logon* na interface gráfica. Será necessário ter conhecimento da senha do superusuário *root* ou estar devidamente cadastrado no sistema.

Não existe dificuldade para executar esta tarefa uma vez que haverá uma janela pedindo o nome do usuário e a senha, basta entrar com os dados.

Tarefa 4 - Identificar qual interface gráfica está instalada na estação em que está utilizando. (Dica: pode ser visto no momento do *logon*).

9.1.2.3 Interface texto

O resultado da tarefa 1 mostra que existem seis interfaces no modo texto (F1 a F6).

Tarefa 5 – Fazer *logon* numa das interfaces texto, com os dados do superusuário *root* e em outra com os dados de usuário comum. Para navegar entre as sessões utilizar as teclas <CTRL> + <ALT> + <F1>; <CTRL> + <ALT> + <F2>; e assim por diante.

Observar que após a autenticação do usuário o sistema fica disponível

para que o usuário possa trabalhar. Descrever qual e a diferença do *PROMPT* (sinal de pronto) disponibilizado para o superusuário *root* e o usuário comum.

Tarefa 6 – Fazer o *logout* das sessões abertas na questão anterior. Os comandos para fechar o *logon* são: *exit* e *logout*. Também pode ser utilizado o atalho pressionando juntas as teclas <CTRL> + <D>.

9.1.3 Desligando e Reiniciando

Um dos grandes problemas da informática são os usuários que desligam os computadores diretamente no botão *power*. Esta prática traz conseqüências indesejáveis ao sistema operacional. As tarefas abaixo indicam o caminho correto para desligar e reiniciar com segurança o computador.

Os comandos novos utilizados nestas atividades são: *shutdown*, *halt*, *poweroff* e *reboot*.

9.1.3.1 Desligando

No Linux existem várias formas (comandos) para desligar o sistema operacional. Os comandos desativam o sistema e a ação efetiva de desligar será feita pressionando o botão *power* após aparecer a mensagem: “*Power down*”.
Obs.: Para máquinas com fonte ATX o desligamento é automático, não sendo necessário pressionar o botão *power*.

Tarefa 1 – Desligando pela interface texto. Basicamente são três os comandos utilizados para solicitar a desativação do sistema: *shutdown*, *halt* e *poweroff* (dependendo da distribuição alguns comandos podem não

funcionar ou será necessário digitar o caminho completo do comando como: `/sbin/shutdown -h now`). (Observar que no Linux é utilizada a barra normal `"/` e não a barra invertida `"\"`).

Tentar desativar o sistema pela interface texto, utilizando o usuário comum e executando o comando `halt`.

Executar novamente o mesmo comando na interface texto utilizando o superusuário `root`. Com qual dos usuários houve a efetiva desativação do sistema? (Obs.: Em algumas distribuições funcionará o comando também para usuário comum, isto dependerá das configurações. O correto é funcionar somente para o usuário `root`).

Fazer o mesmo teste com os comandos: `shutdown -h now` e `poweroff`.

Tarefa 2 – Desligando pela interface gráfica. Procurar por uma opção “Fechar Sessão” ou algo semelhante. Normalmente irá abrir uma janela que terá a opção para desligar o computador

9.1.3.2 Reiniciando

Reiniciar diretamente pelo botão `reset` pode acarretar ao sistema os mesmos transtornos do desligamento feito diretamente pelo botão `power`.

Para reiniciar existem basicamente dois comandos: `reboot` e `shutdown`.

Tarefa 3 – Pela interface texto. Testar o comando `reboot` com o usuário comum e com o superusuário `root` e descrever o resultado. Com qual dos usuários houve efetivamente a reinicialização do sistema? (da mesma forma que

para desligar, em algumas distribuições será possível executar como usuário comum, mas o correto é funcionar somente para o superusuário *root*).

Fazer o mesmo teste com o comando: `shutdown -r now`

Tarefa 4 – Reiniciar o sistema pela interface gráfica. Mesmo caminho da tarefa 2.

Tarefa 5 - Para reiniciar o sistema existe mais uma forma que é pelo uso das teclas <CTRL>+<ALT>+, que pode ser feito estando tanto na interface texto como na interface gráfica.

Reiniciar o sistema utilizando as teclas <CTRL>+<ALT>+ logado como usuário comum e como superusuário *root* e veja se existe alguma diferença. Observar que também existe diferença quando executado estando na interface gráfica e na interface texto.

9.1.4 Primeiros comandos.

Os comandos apresentados nesta fase são os necessários para começar a conhecer o Linux. Boas referências são (TIBET, 2001) e (SILVA, 2005).

9.1.4.1 Comando `pwd`

Tarefa 1 – O comando `pwd` mostra o diretório atual. Executar o comando na sessão logado como superusuário *root* e na sessão logado como usuário comum. Observar a diferença entre os resultados.

Cada usuário está a princípio em seu *home directory*, que é um local em disco reservado para ele. Este diretório está por padrão abaixo do diretório

/home, para usuários comuns e em */root* para o superusuário *root*.

9.1.4.2 Comando **ls**

Tarefa 2 – O comando `ls` lista o conteúdo de diretórios (arquivos e diretórios).

Executar o comando `ls` e `ls -l` e observar os resultados.

9.1.4.3 Comando **cd**

Para complementar o entendimento sobre a navegação entre os diretórios, abaixo uma breve explanação sobre caminho absoluto e caminho relativo:

Quando um caminho de diretório iniciar com a barra `"/`, o sistema entende que este caminho inicia do diretório raiz (caminho absoluto). Quando um caminho inicia sem a barra, o sistema entende que o ponto de partida é o diretório atual (caminho relativo).

Exemplo:

- `cd /bin` (caminho absoluto para o diretório `bin`, neste caso o comando pode ser executado estando posicionado em qualquer diretório que o sistema irá se posicionar no diretório `/bin` sem erro, pois iniciará sua busca a partir do diretório raiz `"/`);

- `cd bin` (caminho relativo para o diretório `bin`, neste caso é necessário estar posicionado no diretório raiz `"/`, pois o sistema partirá a busca pelo diretório solicitado a partir do diretório atual).

Tarefa 3 – O comando `cd` é utilizado para navegar entre os diretórios. A sintaxe deste comando é: `cd /caminho_diretório_destino`

Executar a seqüência de comandos abaixo e observar, pela execução do comando `pwd`, após cada comando, a alteração do diretório atual.

```
cd /usr/bin          cd /etc          cd /          cd
```

9.1.4.4 Comando `su`

Tarefa 4 – Até agora para executar algum comando como o superusuário *root*, foi solicitado que fosse feito um *logon* com este usuário. O comando `su` facilita o uso dos poderes deste superusuário sem necessidade de abrir uma nova sessão.

- Logado como usuário comum, executar o comando `su`. Entre com a senha do superusuário *root* e observar o que acontece.
- Testar executando algum dos comandos anteriormente apresentados que só funcionaram para o superusuário *root*. Veja também a alteração do *PROMPT*.
- Para voltar ao usuário comum executar `exit`.

Obs.: Devido a possibilidade de cometer erros catastróficos como superusuário *root*, sempre usar o sistema como usuário normal e tornar-se *root* apenas temporariamente e somente para executar as tarefas de administração do sistema.

9.1.5 Comando `clear`

Tarefa 5 – Este é um comando simples, utilizado para limpar a tela. Executar o comando `clear` e observar o resultado. Este comando facilita a visualização na

execução de novos comandos, pois somente fica na tela o resultado do comando posterior ao `clear`.

9.1.6 Árvore de diretórios

Praticamente todas as distribuições do Linux usam uma estrutura de diretório semelhante, e os fornecedores de distribuições Linux geralmente têm concordado com a atribuição de nomes e com a localização de arquivos e diretórios importantes no Linux.

Para desempenhar qualquer atividade de administração do sistema é necessário saber como está subdividido o sistema. As tarefas abaixo apresentam alguns dos principais diretórios do sistema de arquivos do Linux. Algumas referências para estas tarefas são (TIBET, 2001) e (BALL, 2004).

Tarefa 1 – Para visualizar a árvore principal de diretórios, executar na seqüência os comandos:

```
cd /      (muda para o diretório raiz, que é representado pela barra "/")  
ls -l
```

Tarefa 2 – Pesquisar e descrever sobre o que compõe cada diretório da árvore principal de diretórios do Linux, visualizado na tarefa anterior.

Tarefa 3 – Diretórios `/bin` e `/sbin`. Nestes diretórios são encontrados os comandos essenciais para a linha de comando utilizado por usuários comuns e pelo superusuário *root*.

- Listar o conteúdo destes diretórios utilizando os comandos:

```
ls /bin e ls /sbin.
```

- Verificar o conteúdo de cada diretório e reconhecer em qual diretório estão os comandos que foram apresentados até agora como: `shutdown`, `pwd`, `ls`, `poweroff`, `halt` e `su`.

- Em qual dos dois diretórios estão os principais comandos da linha de comando e que podem ser executados tanto pelo *root* quanto por usuários comuns?

- Em qual dos dois diretórios estão os comandos para administração do sistema e que na maioria somente o usuário *root* pode executar?

Tarefa 4 – Abaixo de qual diretório estão os diretórios *home* dos usuários cadastrados no sistema? Utilizando o comando `ls /home` listar este diretório e citar quais usuários cadastrados no sistema possuem diretório *home*.

Tarefa 5 – Em qual diretório está o diretório *home* do superusuário *root*? Utilizando o comando `ls` listar o conteúdo do diretório *home* do superusuário *root*. Observar que somente o *root* tem acesso a este diretório.

Tarefa 6 – Diretório `/etc`. Neste diretório estão os principais arquivos de configuração do sistema.

- Utilizando o comando `ls /etc` visualizar o conteúdo deste diretório.

- Utilizando o comando `cat /etc/passwd` visualizar o arquivo dos usuários cadastrados no sistema.

- Utilizando o comando `cat /etc/group` visualizar os grupos cadastrados no sistema.

Tarefa 7 – Da mesma forma como foi feito nas tarefas anteriores, utilizando o comando `ls`, visualizar o conteúdo dos demais diretórios do sistema de arquivos.

9.1.7 Características do Linux

O Linux possui algumas particularidades. Como o uso da tecla <TAB> para completar comandos, a existência de um histórico dos comandos executados, visualização dos resultados dos comandos que ultrapassam o limite da tela e *case sensitive* (diferencia letras maiúsculas e minúsculas).

9.1.7.1 Uso do <TAB> para completar comandos

O Linux tem uma característica de autocompletar comandos que é acionada pela tecla <TAB>.

Tarefa 1 – Testar com os comandos já conhecidos. Digitar apenas as primeiras letras como “pw” e teclar <TAB>. Observar o resultado. Testar com os outros comandos. (Obs.: Enquanto existirem coincidências será apresentado todos os comandos que a possuírem).

Esta característica do Linux é muito útil na utilização de comandos com caminhos extensos ou para encontrar comandos que foram esquecidos.

9.1.7.2 Uso do Histórico dos Comandos

Esta característica do Linux é acionada com o uso das setas. Cada usuário possui seu histórico que não é apagado quando o sistema é reiniciado ou desligado.

Tarefa 2 – Executar novamente os comandos na tarefa anterior sem digitá-los e sim utilizando o histórico.

9.1.7.3 Uso das teclas <SHIFT> + <Page Up> e <Page Down>

Estas teclas são utilizadas para navegar pela tela. Servem para visualizar resultados de comandos que ultrapassam os limites da tela e até mesmo resultados de comandos anteriores.

Tarefa 3 – No modo texto, executar a seqüência dos comandos abaixo e observar a utilidade das teclas <SHIFT> + <Page Up> e <SHIFT> + <Page Down>.

```
cd /home          ls -l
cd /              ls -l
cd                ls -la
```

É bem provável que após executar todos os comandos acima, o resultado do primeiro comando `ls -l` (executado para o diretório */home*) não esteja mais aparecendo na tela. Visualizar novamente o resultado deste comando sem executar o comando, apenas navegando pela tela utilizando as teclas <SHIFT> + <Page Up> e <SHIFT> + <Page Down>.

9.1.7.4 *Case Sensitive* - Diferenciação de letras maiúsculas e minúsculas

O Linux faz diferenciação das letras maiúsculas das minúsculas.

Tarefa 4 - Executar os comandos já conhecidos escrevendo-os todo ou em parte em maiúsculo e depois totalmente em minúsculo e observar o resultado.

Exemplo: Pwd, Ls, Cd.

9.2 Segunda Fase – Básico

Esta fase traz os comandos necessários para execução das operações básicas necessárias para execução das tarefas rotineiras. Entre elas: a manipulação de arquivos e diretórios; como obter ajuda; criação de atalhos; editores e utilitários de texto; permissões de arquivos e diretórios; compactação e descompactação de arquivos e diretórios e data e hora.

As referências que podem ser utilizadas são (TIBET, 2001), (SILVA, 2005), (VEIGA, 2004), (NEVES, 2005) e (ARAUJO, 2001).

9.2.1 Como Obter Ajuda

Por mais que se conheça um sistema operacional, vez ou outra será necessária buscar informações adicionais sobre a funcionalidade de um determinado comando.

Existem várias maneiras de se conseguir ajuda no Linux. A seqüência de atividades abaixo auxilia no processo do aprendizado de como utilizar estas fontes de informação. Uma boa referência sobre como obter ajuda pode ser encontrada em (SILVA, 2005).

9.2.1.1 Páginas de manual (man)

O Linux é um sistema de documentação automática, com páginas de manual acessíveis por meio do comando `man`. As páginas de manual trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento das opções. Esta documentação acompanha quase todos os programas Linux.

Dica: Abrir duas sessões em modo texto, numa sessão executar o comando `man` e na outra testar os comandos.

Tarefa 1 – Utilizando o manual (comando `man`), descobrir como executar o comando `cd` para as solicitações abaixo: (`man cd`).

a - Se posicionar no diretório raiz. (`cd /`).

b - Voltar ao diretório *home* sem utilizar o caminho do diretório.

c - Voltar ao diretório anterior (no caso o raiz) sem usar o comando utilizado no item a. Existe um parâmetro para o comando `cd` que retorna ao diretório anterior ao atual.

d – Se posicionar no diretório `/usr/sbin`.

e – Ir ao diretório `/usr/bin` sem utilizar o caminho completo do diretório, basta no próprio comando `cd`, voltar ao diretório `usr` e depois entrar no diretório `bin`, com apenas um comando. Para voltar um diretório pode ser usado dois pontos `..`, (`cd ..`).

Obs.: Para verificar o diretório atual sempre utilizar o comando `pwd`. Observar também as explicações sobre caminho absoluto e relativo, já apresentadas. Para sair do manual basta pressionar a tecla `q` e para obter mais informações sobre o `man` basta executar: `man man`.

9.2.1.2 *Info Pages*

É semelhante às páginas de manual, mas contém atalhos para outras páginas. A sintaxe do comando é: `info [comando/programa]`.

Tarefa 2 – Pesquisar mais sobre o comando `ls`, já apresentado, utilizando o

comando `info` e executar as solicitações abaixo: (`info ls`).

a - Listar o conteúdo do diretório corrente por ordem de data e hora da última modificação, listando um arquivo por linha.

b - Listar todos os arquivos do diretório *home*, inclusive os arquivos ocultos (iniciados por ponto "."). (Antes executar o comando `cd` para retornar ao *home directory*).

c - Listar o conteúdo de todos os diretórios e subdiretórios a partir do diretório raiz. (para ir ao diretório raiz, executar o comando: `cd /`).

d - Descobrir e testar mais parâmetros para o comando `ls`.

Obs.: Para obter mais informações sobre como utilizar o comando `info`, executar `man info` ou simplesmente `info`.

9.2.1.3 *Help on line*

É uma ajuda rápida, que mostra quais as opções possíveis para o programa/comando em questão. Sintaxe: `[comando] --help`

Tarefa 3 – Visualizar quais são os parâmetros possíveis para o comando `ls` utilizando o *help on line*. (Visualizar o resultado do comando com as teclas <SHIFT> + <Page Up> e <SHIFT> + <Page Down>).

9.2.1.4 *Help*

Também é uma ajuda rápida, mas somente para os comandos internos do interpretador de comandos (os interpretadores de comandos ou *Shell* serão vistos mais a frente). Sintaxe: `help [comando]`

Tarefa 4 – Visualizar as opções possíveis para o comando `exit` e `echo` utilizando o comando `help`.

9.2.1.5 Documentos *HOWTO*'s

São documentos que explicam como fazer determinada tarefa. Caso tenha sido instalado junto com a distribuição, são encontrados no diretório `/usr/doc/how-to`.

Também são encontrados na Internet nos caminhos abaixo:

<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>

<http://www.tldp.org/projetos/howto/traduzidos.php>

Tarefa 5 – Capturar e estudar pelo menos um *HOWTO*.

9.2.1.6 Documentação dos aplicativos

Semelhante aos *HOWTO*. É a documentação dos programas que geralmente são instalados juntos no momento da instalação do programa e normalmente estão no caminho `/usr/doc/[programa]`.

Tarefa 6 – Procurar em `/usr/doc` quais documentações estão instaladas, escolher uma delas e estudar seu conteúdo. Somente se existir alguma documentação instalada.

9.2.1.7 Comandos: locate, slocate, which, apropos e whatis

Tarefa 7 – Pesquisar quais são as semelhanças e diferenças dos programas locate, slocate, which, apropos e whatis. Descrever para que servem e testar todos eles com os comandos e arquivos já conhecidos como: reboot, halt, logout, poweroff, shutdown, ls, cd e pwd.

Dica: Utilizar os comandos de ajuda conhecidos como o man, help e info para estudar os comandos da tarefa 7, acima.

9.2.1.8 Internet / livros

Estas são as outras fontes de informação, externas ao sistema.

Tarefa 8 – Acessar a página <http://www.google.com/linux> e pesquisar sobre *software* livre ou sobre o porque do pinguim ser o símbolo do Linux.

9.2.2 Manipulação de Arquivos e Diretórios

O grupo de tarefas nesta etapa estão focadas na manipulação dos arquivos e diretórios, como criar, excluir, mover, copiar e ler.

Como no item anterior foi apresentado como encontrar ajuda, a partir deste ponto, as informações sobre os comandos serão reduzidas. Será, portanto, necessária pesquisa adicional para completar as tarefas, como (SILVA, 2005), (ARAÚJO, 2001), (NEVES, 2005) e (VEIGA, 2004).

Para a execução das tarefas ir para o diretório *home*. Basta executar o comando `cd`. Observar que para executar estas tarefas, pode ser utilizado tanto o caminho absoluto como o caminho relativo, já tratado anteriormente.

9.2.2.1 Extensão dos arquivos

Tarefa 1 – Pesquisar e responder as questões abaixo referentes a extensões dos arquivos no Linux:

a – A extensão dos arquivos é necessário no Linux?

b – Para que é utilizada a extensão nos arquivos no Linux?

c – Descrever para que tipo de arquivo é utilizada as extensões: `.sh`, `.txt`, `.zip`, `.gz`, `.log`, `.html` e `.conf`.

9.2.2.2 Arquivos Ocultos

Tarefa 2 – O que caracteriza um arquivo oculto no Linux e qual parâmetro é necessário acrescentar ao comando `ls` para visualizá-los? Listar os arquivos ocultos do diretório *home* do usuário logado.

9.2.2.3 Conectores

Os conectores no Linux são utilizados para facilitar a execução dos comandos. São eles: *pipe* [`|`]; redirecionamento [`>`]; concatenação [`>>`].

A utilização dos conectores será vista juntamente com as próximas atividades.

9.2.2.4 Manipulação de Diretórios

Os comandos novos para esta atividade são: `mkdir`, `rmdir`, `rm`, `cp` e `mv`. Será necessário buscar informações da sintaxe e parâmetros dos comandos

pelo `man` ou outro meio de ajuda/informação.

Atividade 3 – Usando o comando `mkdir`, criar um diretório chamado **teste** dentro do diretório *home*. Visualizar o diretório recém criado utilizando o comando `ls`.

Atividade 4 – Novamente com o comando `mkdir` criar outro diretório chamado **teste1**, também dentro do diretório *home* do usuário.

Atividade 5 – Com o comando `cd` se posicionar no diretório **teste** e utilizando o comando `mv` mover o diretório **teste1** para dentro do diretório **teste**.

Atividade 6 – Fazer uma cópia do diretório **teste1** para o diretório anterior, ou seja, o diretório *home* do usuário. Utilizar o comando `cp` para fazer a cópia.

Atividade 7 – Remover o diretório **teste1** recém copiado do diretório *home*, utilizando o comando `rmdir`.

Atividade 8 – Remover também o diretório **teste** utilizando o comando `rmdir`. Observar que o diretório **teste** não está vazio, portanto a forma para exclusão é diferenciada. Caso não consiga com o comando `rmdir`, tentar usar o comando `rm`. Pesquisar sobre os dois comandos.

9.2.2.5 Manipulação de Arquivos

Os comandos novos para esta atividade são: `touch`, `cat`, `tail`, `head`, `more`, `less` e `rename`. Os comandos auxiliares e já vistos são: `cd`, `ls`, `pwd`, `mkdir`, `rm`, `cp` e `mv`.

Atividade 9 – Usando o comando `mkdir`, criar um diretório chamado **teste** dentro do diretório *home*. Visualizar o diretório recém criado utilizando o comando `ls`.

Atividade 10 – Com o comando `cp`, já conhecido, copiar o arquivo `/etc/passwd` e o arquivo `/etc/group` para dentro do arquivo recém criado. Antes utilizar o comando `cd` para se posicionar no diretório **teste**.

Atividade 11 – Visualizar o conteúdo do arquivo `passwd` e `group` utilizando o comando `cat`. Veja também com que opção é possível visualizar o conteúdo do arquivo com as linhas numeradas.

Atividade 12 – Criar um arquivo vazio com o nome de `arq` utilizando o comando `touch`. O comando `touch` além de criar arquivos vazios tem outra funcionalidade. Pesquisar sobre esta outra funcionalidade.

Atividade 13 – Utilizando o comando `cat`, copiar o conteúdo do arquivo `passwd` para o arquivo `arq`. Neste caso será necessário utilizar o redirecionamento, sinal de maior `>`. (`cat passwd > arq`). Visualizar o conteúdo do arquivo `arq` utilizando o comando `cat`.

Atividade 14 – Novamente utilizando o comando `cat` concatenar o conteúdo do arquivo `group` no arquivo `arq`. Neste caso será necessário utilizar o símbolo para concatenação `>>`. Visualizar o conteúdo do arquivo `arq` utilizando o comando `cat`. O arquivo `arq` deverá possuir o conteúdo dos dois arquivos, `passwd` e `group`.

Atividade 15 – Visualizar o conteúdo do arquivo `arq` utilizando os comandos `more` e `less`. Pesquisar sobre os comandos e observar quais as diferenças entre eles. Os comandos `more` e `less` podem ser utilizados em combinação com

outros comandos. Neste caso será necessário fazer uso do conector *pipe* [|], que direciona a saída de um comando para a entrada de outro. Como exemplo: `cat arq | more`. Testar também com o comando `less`.

Atividade 16 – Visualizar somente as últimas 05 linhas do arquivo `passwd` utilizando o comando `tail` e as 05 primeiras linhas utilizando o comando `head`.

Atividade 17 – Utilizando o comando `mv` mover o arquivo `arq` para o diretório *home* do usuário.

Atividade 18 – Utilizando o comando `mv` alterar o nome do arquivo `arq` para `arq1`.

Atividade 19 – Utilizando o comando `rename` alterar o nome do arquivo `arq1` para `arqteste`.

Atividade 20 – Utilizando o comando `rm` remover todos os arquivos e pastas criadas nas atividades acima.

9.2.3 Permissões de acesso a arquivos e diretórios

Todos os arquivos e diretórios no Linux possuem permissões de acesso que são visualizadas pelo comando `ls -l`. A opção (`-l`) mostra um arquivo por linha no formato longo, mostrando as permissões, dono, grupo, data da modificação e tamanho.

As tarefas abaixo têm por objetivo apresentar como decifrar e alterar as permissões dos arquivos e diretório no Linux.

9.2.3.1 Reconhecendo as permissões

Para as tarefas abaixo serão utilizados apenas comandos já conhecidos.

Tarefa 1 – Utilizando o comando `cd` se posicionar no diretório `/home` do usuário logado. Executar o comando `ls -l` e observar os dados apresentados dos arquivos e diretórios. Notar que o primeiro dígito da linha é diferente podendo ser `(-)` ou `(d)`, o que indica esse dígito?

Tarefa 2 – Com o comando `touch` criar um arquivo vazio com o nome que desejar.

Tarefa 3 – Com o comando `ls -l`, visualizar as permissões do arquivo recém criado. Observar que nas permissões existem três conjuntos de três dígitos cada. Pesquisar e responder o que indica cada letra do conjunto de dígitos:

r -

w -

x -

Tarefa 4 – Descrever para que serve cada um dos três conjuntos de dígitos (`rwX`).

Primeiro grupo de três dígitos `rwX` –

Segundo grupo de três dígitos `rwX` –

Terceiro grupo de três dígitos `rwX` –

Tarefa 5 – Observar o exemplo abaixo e escrever o que será permitido e negado para o dono do arquivo, para o grupo de dono do arquivo e para os outros.

```
-rw-rw-r-- 2 root root 2159 Set 5 14:30 teste
```

- Permissão para o dono do arquivo =
- Permissão para o grupo do dono do arquivo =
- Permissão para os outros =

Tarefa 6 – Descrever o que é permitido com as tríades abaixo:

`rxw =`

`rw- =`

`r-- =`

9.2.3.2 Alteração das permissões dos arquivos e diretórios

Os comando novos para estas tarefas são: `chmod`, `chgrp` e `chown`.

Para visualizar as permissões utilizar o comando `ls -l`.

Observar que para alterar as permissões de arquivos e diretórios é necessário possuir os poderes do superusuário *root*. Portanto executar antes o comando `su` para alterar os ID's de usuário e de grupo para os do superusuário *root*.

Tarefa 7 – Utilizando o comando `chmod`, alterar a permissão do arquivo criado na tarefa 2 para que todos (dono, grupo e outros) tenham acesso de leitura, escrita e execução.

Tarefa 8 – Alterar novamente as permissões do arquivo retirando a permissão de execução de todos (dono, grupo e outros) e retirando também a permissão de escrita para o grupo e para outros.

Tarefa 9 – Utilizando o comando `chgrp` alterar o grupo do arquivo para *root*.

Tarefa 10 – Utilizando o comando `chown` alterar o dono do arquivo para *root*.

Tarefa 11 – Alterar novamente o dono e o grupo para o usuário que o criou.

Tarefa 12 – As permissões também podem ser indicadas na forma numérica.

Sendo: **r** = 4; **w** = 2 e **x** = 1 (4 + 2 + 1 = 7).

--- = 0 + 0 + 0 -> 0

r-- = 4 + 0 + 0 -> 4

rw- = 4 + 2 + 0 -> 6

rwX = 4 + 2 + 1 -> 7

Com as informações acima e após pesquisar sobre as permissões de acesso, indique em octal as permissões abaixo:

r--r--r-- = 444 (exemplo)

rw-r--r-- =

r-xr-xr-x =

rwXr-xr-x =

r-xr-xr-x =

rw-rw-r-- =

rwXrwXrwX =

9.2.4 Criando Atalhos

Em Linux existem duas formas de atalhos: O *hard link* (ligação fixa) e o *symbolic link* (ligação simbólica).

Para melhor compreensão sobre os atalhos, a seguir temos uma pequena explicação sobre o sistema de arquivos do Linux:

O sistema de arquivos utilizado pelo Linux cria um número para cada pedaço no disco que é dado o nome de *inode*. De forma simples ele associa um número de *inode* a um nome de arquivo o que permite que um arquivo possua diversos nomes. Os nomes poderão estar em diretórios diferentes e os diversos nomes de um mesmo arquivo são chamados de *links*.

Arquivos de qualquer tipo são armazenados nos *inodes* e os nomes dos arquivos indicam qual *inode* o nome se refere. Para visualizar o número do *inode* de um arquivo utilizar o comando `ls` com a opção `-i`. Por exemplo, para visualizar o *inode* do arquivo `/etc/passwd` executar: `ls -i /etc/passwd`. O número que aparece antes do nome do arquivo é o *inode*.

Tarefa 1 – Descreva o que é: *hard link* e *symbolic link*.

9.2.4.1 *Hard Link*

Mais de um arquivo pode indicar o mesmo *inode*. Isso é conseguido através da criação de um *hard link* (ligação fixa). Quando uma ligação fixa é criada, caso o arquivo seja removido, apenas o nome será retirado da lista já que outro nome está apontando para o mesmo *inode*.

O comando novo para estas tarefas é: `ln`. Os demais comandos já foram vistos.

Tarefa 2 – Criação de um *hard link*.

- Utilizando o comando `mkdir` criar um diretório chamado **atalho** dentro do diretório *home* do usuário. (antes se posicionar no diretório *home* utilizando o

comando `cd`).

- Utilizando o comando `cp` copiar o arquivo `/etc/passwd` para dentro do diretório **atalho**. Visualizar a cópia do arquivo com o comando `ls -l ./atalho`.

- Utilizando o comando `ln` e posicionado no diretório *home* criar um *hard link* com nome `hl_passwd` apontando para o arquivo *passwd* copiado para o diretório **atalho**.

- Pelo comando `ls -l` e `ls -l ./atalho` verificar a diferença deste *link* recém criado para os demais arquivos. Observar o tamanho do *link* em relação ao arquivo original.

- Execute `cat hl_passwd` e `cat ./atalho/passwd` e observar o resultado.

- Utilizando o comando `ls -li` visualizar o *inode* do *link* `hl_passwd` e do arquivo `./atalho/passwd`.

- Utilizando o comando `rm` excluir o arquivo *passwd* que está dentro do diretório **atalho** (`rm ./atalho/passwd`).

- Pelo comando `ls -l` observar se houve alterações no *link* `hl_passwd`.

- Executar novamente o comando `cat hl_passwd`.

- Utilizando o comando `cp` copiar novamente o arquivo *passwd* para dentro do diretório **atalho** e observar se há alterações pelos comandos `ls -l` e `cat hl_passwd`.

- Visualizar novamente o *inode* dos dois arquivos utilizando o comando `ls -li`.

9.2.4.2 *Symbolic Link*

Diferente da ligação fixa, a ligação simbólica aponta para o nome do arquivo original. Caso o arquivo original seja excluído, o nome do *link* simbólico continua na lista, mas apontando para um arquivo que não existe mais.

Tarefa 3 – Criação de um *symbolic link*.

- Utilizando o comando `mkdir` criar um diretório chamado **atalho1** dentro do diretório *home* do usuário. (antes se posicionar no diretório *home* utilizando o comando `cd`).
- Utilizando o comando `cp` copiar o arquivo `/etc/passwd` para dentro do diretório **atalho1**. Visualizar a cópia do arquivo com o comando `ls -l ./atalho1`.
- Utilizando o comando `ln` e posicionado no diretório *home* criar um *symbolic link* com nome `sl_passwd` apontando para o arquivo `passwd` copiado para o diretório **atalho1**.
- Pelo comando `ls -l` e `ls -l ./atalho1` verificar a diferença deste *link* recém criado para os demais arquivos. Observar o tamanho do *link* em relação ao arquivo original.
- Executar `cat sl_passwd` e `cat ./atalho1/passwd` e observar o resultado.
- Utilizando o comando `ls -i` visualizar o *inode* do *link* `sl_passwd` e do arquivo `./atalho1/passwd`.
- Utilizando o comando `rm` excluir o arquivo `passwd` que está dentro do diretório **atalho1** (`rm ./atalho1/passwd`).

- Pelo comando `ls -l` observar se houve alterações no *link* `sl_passwd`.
- Executar novamente o comando `cat sl_passwd`.
- Utilizando o comando `cp` copiar novamente o arquivo `passwd` para dentro do diretório **atalho1** e observar se há alterações pelos comandos `ls -l` e `cat sl_passwd`.
- Visualizar novamente o *inode* dos dois arquivos.

Tarefa 4 – Descreva quais as diferenças do *symbolic link* e do *hard link*.

Tarefa 5 – No Linux existe também a possibilidade de criação de *links* simbólicos para diretórios inteiros. Para isto é utilizado o comando `ln -s`. Pesquisar e testar o comando `ln -s`.

9.2.5 Editores de texto

A quantidade de editores de texto instalados no Linux depende da distribuição que estiver sendo utilizada e da quantidade de pacotes de programas instalados.

Sem os editores de texto seria impossível executar as tarefas de administração do sistema que requerem na sua grande maioria alterações em arquivos texto. A preferência pelo domínio de pelo menos um editor de texto baseado em console se deve ao fato de que muitos editores mais robustos da interface gráfica não funcionaram quando o Linux for iniciado em modo monousuário ou de manutenção. Outro motivo é porque muitos servidores não possuem interface gráfica, sendo impossível gerenciar tais servidores sem o domínio da interface modo texto.

O editor de texto mais conhecido é o Vim, que é uma versão melhorada

do `vi`. O Vim também pode ser chamado de `vi`, pois tem um *link* simbólico chamado `vi`.

Outro editor de texto também bastante conhecido é o `emacs`. O `emacs` possui funcionalidades que vão além de um editor de texto, podendo ser usado também para escrever e compilar programas, atuar como um diário eletrônico, e algumas outras funções. Tanto o editor Vim como o editor `emacs` estão presentes por padrão em quase todas as distribuições Linux.

Outros editores encontrados no Linux, dependendo da distribuição são: `pico`, `joe`, `mcedit` (`mc`), `elvis`, `jed`, `gedit`, `gxedit`, `sed`, `ed` e `ae`, entre outros.

9.2.5.1 Editor de texto Vim

Para entender como funciona o editor de texto Vim, verificar em (INFOWESTER, 2003). Este artigo traz de forma detalhada todo o funcionamento deste poderoso editor de texto. Caso queira pode ser buscado informações sobre o editor de texto Vim em livros, como (ARAUJO, 2001), (STATO FILHO, 2004), (JANG, 2003), (TIBET, 2001), (BALL, 2004) e (SHAH, 2003) e no próprio manual do Linux (`man vi`). Praticamente todos os livros sobre Linux trazem informações sobre o editor de texto Vim.

Tarefa 1 - Utilizando o comando `cd` se posicionar no diretório *home* do usuário logado.

Tarefa 2 – Utilizando o editor de texto Vim, criar um arquivo chamado `poesia.txt` (`vi poesia.txt`).

Tarefa 3 – Observar que ao executar o editor de texto Vim, o mesmo inicia em

modo de comando. Para começar a digitar o texto será necessário alterar para o modo de inserção pressionando no teclado a letra "i". Observar pela informação "INSERT" no rodapé do arquivo. Após entrar em modo de inserção digitar o texto abaixo:

Educação é uma coisa admirável.

Mas é sempre bom lembrar,

De tempos em tempos, que nada daquilo

Que realmente vale a pena saber

Pode ser ensinado.

Oscar Wilde

Tarefa 4 – Após digitar o texto acima, salvar o arquivo sem sair. Pressionar a tecla <ESC> para entrar no modo comando e digitar o comando para salvar (:w).

Tarefa 5 – Voltar ao modo de inserção pressionando a tecla "i". Digitar o texto abaixo após o texto digitado na tarefa 3. Deixar três linhas em branco entre um texto e outro.

A mudança não ocorre quanto a

Exigimos de nós, mas sim quanto nos

Conscientizamos daquilo que não

Está dando certo.

Shakti Gawain

Tarefa 6 – Após digitar o texto acima, sair do arquivo salvando. Pressionar a

tecla <ESC> para entrar no modo comando e digitar o comando para salvar e sair (:wq).

Tarefa 7 – Utilizando o editor de texto Vim entrar novamente no arquivo poesia.txt. Entrar no modo de inserção pressionando a tecla "i". Digitar qualquer coisa no meio do texto e sair do arquivo sem salvar. Pressionar a tecla <ESC> para entrar no modo comando e digitar o comando para sair sem salvar (:q!). Utilizando o comando cat poesias.txt, observar que o que foi digitado não foi gravado no arquivo.

Tarefa 8 – Utilizando o editor de texto Vim entrar novamente no arquivo poesia.txt e incluir no final do arquivo o texto abaixo, deixando três linhas em branco entre um texto e outro:

Às vezes é preciso estragar um
Pouquinho o quadro para terminá-lo.
Eugène Delacroix

Tarefa 9 – Após digitar o texto indicado na tarefa anterior, salvar o arquivo com o nome de frases.txt. (:wq frases.txt).

Tarefa 10 – Utilizando o comando cat, visualizar o conteúdo do arquivo poesias.txt e do arquivo frases.txt. Observar a diferença entre os dois.

Tarefa 11 - Utilizando o editor de texto Vim entrar novamente no arquivo poesia.txt e excluir uma das linhas em branco entre o primeiro texto e o segundo texto.

Tarefa 12 – Pesquisar sobre os demais comandos do editor de texto Vim e testá-

los nos arquivos `poesia.txt` e `frases.txt` já criados.

Tarefa 13 – Pesquisar sobre os outros editores de texto informados acima ou existentes na distribuição que está sendo utilizada e testá-los como foi feito com o editor de texto Vim.

9.2.6 Utilitários de texto

No Linux existe uma grande variedade de arquivos do tipo texto, como os *logs*, os *Shell Scripts*, informativos sobre o sistema e os códigos-fontes entre muitos outros. De igual modo, no Linux, existe uma variedade de programas que são utilizados para tratamento destes arquivos-texto.

Os comandos novos para estas tarefas são: `grep`, `wc`, `sort`, `cut`, `nl` e `diff`.

Tarefa 1 - Utilizando o comando `grep`, procurar pela ocorrência `"/bin/sh"` no arquivo `/etc/passwd`, visualizando o resultado da ocorrência com o número da linha.

Obs.: O comando `grep` pode ser utilizado também para filtrar saídas de outros comandos. Por exemplo: `ls /etc | grep resolv` (lista o diretório `/etc` com o comando `ls` e direciona a saída do comando `ls` para o comando `grep`, que por sua vez procura pela expressão `resolv`). Para arquivos compactados existe o comando `zgrep`, que funciona exatamente da mesma forma que o comando `grep`.

Tarefa 2 – Utilizando o comando `wc`, descobrir quantas linhas, quantas palavras e quantos *bytes* existem no arquivo `/etc/passwd`.

Tarefa 3 – Criar dois arquivos texto dentro do diretório *home* do usuário logado, com os seguintes conteúdos:

- Primeiro arquivo (criar com o nome de `arq1`):

Mamão

Abóbora

Banana

Chuchu

- Segundo arquivo (criar com o nome de `arq2`):

Trigo

Feijão

Abacate

Soja

Tarefa 7 – Utilizando o comando `sort`, visualizar o conteúdo dos arquivos `arq1` e `arq2` em ordem alfabética.

Tarefa 8 – Utilizando o comando `sort`, mesclar os dois arquivos, `arq1` e `arq2`, no arquivo `arq3`.

Neste caso será necessário utilizar o redirecionamento [`>`].

Tarefa 9 – Utilizando o comando `cut`, extrair apenas a primeira coluna do arquivo `/etc/passwd`. Considerar como limitador para as colunas do arquivo os dois pontos `:"`.

Tarefa 10 – Utilizando o comando `nl` exibir o arquivo `/etc/passwd` com as linhas numeradas.

Tarefa 11 – Utilizando o comando `diff` visualizar as diferenças dos arquivos `arq1` e `arq3`. **Obs.:** para arquivos compactados existe o comando `zdiff`.

9.2.7 Compressão e empacotamento de arquivos e diretórios

O sistema operacional Linux oferece meios bastante eficientes para empacotamento/desempacotamento e compressão/descompressão de arquivos. Por empacotamento entende-se a ação de coletar arquivos e/ou diretórios e colocar todos juntos num arquivo único e por compressão entende-se a ação de reduzir o tamanho de um arquivo. Desempacotamento e descompressão é a ação contrária.

Os comandos novos utilizados nestas tarefas são: `compress`, `tar`, `zip`, `gzip`, `unzip` e `gunzip`.

Tarefa 1 – O comando `tar` serve tanto para empacotar como para desempacotar arquivos, com ou sem opção para compressão. Por exemplo, para empacotar todos os arquivos `.conf` do diretório `/etc`, é utilizado o comando: `tar -cvf conf.tar /etc/*.conf`. Utilizando o comando `tar`, visualizar o conteúdo do arquivo `conf.tar` e desempacotar os arquivos em outro diretório.

Tarefa 2 – O comando `zip` compacta e empacota arquivos para *UNIX*, *DOS*, *Windows*, *Macintosh*, *OS/2*, *Atari*, *Windows NT*, *Minix* e *VMS*. Como para o Linux existem outros comandos com mais recursos como o `tar`, normalmente o

`zip` só é utilizado quando o arquivo será aberto em outro sistema operacional. Para descompactar arquivos compactados pelo `zip` é utilizado o comando `unzip`. Utilizando o comando `zip`, compactar o arquivo `/etc/passwd` no diretório `home` do usuário logado.

Tarefa 3 – O comando `gzip` comprime arquivos. Os arquivos comprimidos pelo `gzip` podem ser descomprimidos pelo próprio comando `gzip` ou pelo comando `gunzip`. Utilizando o comando `gzip`, compactar o arquivo `conf.tar` criado na tarefa 1. Utilizando ainda o comando `gzip`, visualizar o conteúdo do arquivo recém compactado, `conf.tar.gz`. Descomprimir utilizando o comando `gunzip`.

9.2.8 Data e Hora

O comando para visualizar e alterar a data e hora é o `date` e para visualizar um calendário simples é o `cal`.

Tarefa 1 – Utilizando o comando `date` visualizar a data e hora atual.

Tarefa 2 – Existem vários formatos para as saídas do comando `date`. Após verificar no manual do comando execute as tarefas abaixo:

- Visualizar apenas o ano corrente.
- Visualizar apenas a data no formato DD-MM-AAAA
- Visualizar o horário no formato *hh:mm:ss*.
- Visualizar apenas os dois últimos dígitos do ano.
- Visualizar a data sem nenhum separador – DDMMAAAA (Será necessário mesclar os códigos para formatação da data).

Tarefa 3 – Utilizando o comando `cal` visualizar o calendário para o mês atual.

Tarefa 4 – Utilizando o comando `cal` visualizar o calendário para o ano atual.

Direcionar o resultado do comando para dentro de um arquivo chamado `ano`.

Visualizar o conteúdo do arquivo utilizando o comando `cat`.

9.3 Terceira Fase – Intermediário

Esta fase aborda alguns procedimentos mais complexos, como a administração de usuários e grupos; gerenciamento dos processos e do sistema; agendamento de tarefas; criação de apelidos (*alias*) e equivalências entre *Windows/DOS* e *Linux*.

Boas referências para esta fase são (TIBET, 2001), (SILVA, 2005) e (VEIGA, 2004).

9.3.1 Administração de usuários

Os comandos novos utilizados nestas atividades são: `adduser` ou `useradd`, `addgroup`, `passwd`, `newgrp`, `userdel`, `groupdel`, `id`, `usermod`, `logname`, `users`, `groups`, `last`, `w`, `finger`, `who` e `whoami`.

Para a execução das atividades relativas a manipulação de contas e grupos dos usuários, na grande maioria será necessário ter poderes do superusuário *root*. O que pode ser conseguido com a utilização do comando `su`.

9.3.1.1 Criação de contas e grupos de usuários

Os comandos novos utilizados para estas tarefas são: `useradd` ou `adduser`, `passwd`, `addgroup` ou `groupadd`.

Os usuários no *Linux* podem ser criados a partir dos padrões configurados no sistema ou por definições informadas no momento do cadastramento. Para verificar os padrões que estão definidos no sistema basta executar o comando `adduser -D` ou `useradd -D`.

Pesquisar sobre o comando `adduser` ou `useradd` depende da distribuição que estiver sendo utilizada, e executar as tarefas abaixo:

Tarefa 1 – Utilizando o comando `adduser` ou `useradd`, cadastrar um novo usuário no sistema utilizando os padrões definidos no sistema.

Tarefa 2 – Utilizando o comando `passwd` atribuir uma senha para o usuário criado na tarefa anterior.

Tarefa 3 – Verificar com o comando `cat` o conteúdo do arquivo `/etc/passwd` e procurar pelos dados do usuário recém criado. (`cat /etc/passwd`).

Tarefa 4 – Verificar se foi criado o diretório *home* para o novo usuário dentro do diretório */home*.

Tarefa 5 – Utilizando o comando `groupadd` ou `addgroup`, depende da distribuição, criar o grupo **estudo**. Executar o comando `cat /etc/group` e procurar pelo grupo recém criado. Observar que existe um número (ID) associado a este grupo.

Tarefa 6 – Utilizando o comando `adduser` ou `useradd` criar um outro usuário atribuindo a este usuário no próprio comando de criação o grupo **estudo** criado na tarefa anterior. Executar o comando `cat /etc/passwd` e procurar pelo novo usuário. Observar que a quarta informação é o mesmo número associado ao grupo criado na tarefa anterior. Também pode ser observado pelo comando `ls -l /home` que o diretório *home* do usuário possui como grupo o mesmo grupo que foi atribuído ao usuário.

9.3.1.2 Exclusão de contas e grupos de usuários

Os comandos novos utilizados para estas tarefas são: `userdel`, `groupdel`.

Tarefa 7 – Utilizando o comando `userdel`, excluir o usuário criado na tarefa 6. Executar `cat /etc/passwd` e verificar se o usuário foi excluído do arquivo. Verificar no diretório `/home` se o diretório `home` do usuário foi excluído.

Tarefa 8 – Utilizando o comando `userdel`, excluir o usuário criado na tarefa 1, excluindo juntamente os arquivos e diretório `home` deste usuário. Verificar de igual forma que na tarefa anterior se o usuário foi excluído do arquivo `/etc/passwd` e se o diretório `home` do mesmo ainda está no sistema.

Tarefa 9 – Utilizando o comando `groupdel` excluir o grupo estudo criado anteriormente. Visualizar pelo comando `cat /etc/group` se o grupo foi excluído. **Obs.:** Não será possível excluir um grupo enquanto existirem usuários associados a ele.

Tarefa 10 – Verificar as outras opções para o comando `adduser` ou `useradd` e testar criando novos usuários no sistema.

9.3.1.3 Administração de contas e grupos

Os comandos novos utilizados para estas tarefas são: `id`, `usermod`, `logname`, `users`, `groups`, `last`, `w`, `finger`, `who` e `whoami`.

Estas tarefas têm por objetivo apresentar uma série de comandos úteis para a administração dos usuários.

Tarefa 11 – Utilizando o comando `useradd` ou `adduser` criar um novo usuário. Utilizando o comando `passwd` atribuir uma senha a este novo usuário. Visualizar a criação do usuário pelos comandos `cat /etc/passwd` e `ls -l /home`.

Tarefa 12 – Utilizando o comando `addgroup` ou `groupadd` criar um novo grupo denominado **contabilidade**. Visualizar a criação do grupo pelo comando `cat /etc/group`.

Tarefa 13 – Utilizando o comando `usermod` atribuir ao novo usuário o grupo **contabilidade** criado na tarefa anterior. Visualizar quais foram as alterações pelos comandos `cat /etc/passwd` e `cat /etc/group`.

Tarefa 14 – Utilizando o comando `id` visualizar o ID do usuário logado, do usuário criado na tarefa 12 e do superusuário *root*. Verificar as demais opções possíveis do comando `id` para visualização dos ID's dos usuários.

Tarefa 15 – Utilizando o comando `logname` visualizar o nome de acesso do usuário logado. Faça *logon* em outro terminal com o usuário criado na tarefa 12 e executar novamente o comando `logname`.

Tarefa 16 – Utilizando o comando `users` verificar quantos usuários estão logados no sistema.

Tarefa 17 – Utilizando o comando `groups` verificar a quais grupos o usuário logado pertence. Verificar também a quais grupos o usuário criado na tarefa 12 pertence.

Tarefa 18 – Utilizando o comando `last` verificar quais foram os últimos usuários que utilizaram o sistema.

Tarefa 19 – Os comandos `finger`, `w`, `who` e `whoami` trazem informações

sobre quem está logado no sistema. Executar os comandos e observar as informações que cada um deles traz. **Obs.:** Em algumas distribuições não existe o comando `finger`.

9.3.2 Gerenciamento de processos

No Linux, qualquer programa em execução é considerado um processo. Para diferenciar um processo de outro, o sistema atribui um número para cada processo, denominado PID. É através deste identificador que o processo é manipulado.

No Linux, é possível verificar a qualquer momento o que está sendo executado no sistema, quer seja do próprio usuário logado, quer seja de todo o sistema. Para visualização dos processos de todo o sistema é necessário ter poderes do superusuário *root*.

Os comandos novos utilizados para estas tarefas são: `ps`, `pstree`, `top`, `kill` e `killall`.

Tarefa 1 – Utilizando o comando `ps` listar de forma resumida, todos os processos do usuário logado.

Tarefa 2 – Efetuar o *logon* em outro terminal e executar o comando `man ps`.

Tarefa 3 – Voltar ao terminal anterior e executar novamente o comando `ps` e verificar qual é o número do processo do comando executado na tarefa anterior. Solicitar todos os processos do usuário logado ou do *tty* em que está sendo executado o comando.

Tarefa 4 – Utilizando o comando `kill`, finalizar o processo do comando executado na tarefa 2.

Tarefa 5 – Verificar as demais opções para o comando `ps` e resolver as questões abaixo: Antes efetuar *logon* em vários terminais com usuários diferentes e executar várias aplicações como: `vi`, `man` e `more`. Isto fará com que vários processos fiquem rodando no sistema, facilitando a execução das tarefas. Obs.: Os comandos executados nos diferentes terminais têm que ficar executando, por exemplo: Executar `man mkdir` num terminal e deixar na tela do `man`, assim o processo do `man` terá que aparecer quando for solicitado pelo comando `ps`.

- Listar todos os processos do terminal atual.
- Listar todos os processos de todos os `tty`.
- Listar todos os processos do usuário `root` ou de algum outro usuário.
- Listar todos os processos exibindo o nome verdadeiro do comando.

Tarefa 6 – O comando `killall` finaliza um processo pelo nome e não pelo ID. Utilizando o comando `killall` finalizar os processos dos comandos que estão sendo executados nos outros terminais. Utilizando o comando `ps` identificar o nome dos processos ativados pela execução dos comandos indicados na tarefa 7 (`man`, `vi` e `more`).

Tarefa 7 – O comando `pstree` é semelhante ao comando `ps`, mas mostra os processos em forma de árvore. Testar o comando `pstree` com seus diversos parâmetros e observe os resultados.

Tarefa 8 – O comando `top`, exibe em tempo real todos os processos que estão sendo executados no sistema. Executar o comando `top` e observar todas as informações que são exibidas.

Tarefa 9 - Executar novamente o comando `top` exibindo a linha de comando dos processos ao invés de apenas o nome.

9.3.3 Gerenciamento do sistema

Os comandos novos utilizados para estas tarefas são: `free`, `df`, `du`, `uptime`, `mount` e `umount`.

O conjunto de comandos apresentados nesta fase são os principais para o gerenciamento do sistema.

Tarefa 1 – Utilizando o comando `free`, verifique a quantidade de memória livre e utilizada pelo sistema. Execute novamente exibindo a quantidade em *Mbytes* e com totalizador.

Tarefa 2 – O comando `df` exibe a quantidade de espaço utilizado pelos sistemas de arquivos. Utilizando o comando `df` verificar qual espaço utilizado pelos sistemas de arquivos atualmente montados no sistema.

Tarefa 3 – O comando `du` exibe o tamanho de arquivos ou diretórios. Utilizando o comando `du` verificar o tamanho do diretório `/etc`. Obs.: Será necessário utilizar a opção que apresenta um total no final.

Tarefa 4 – Utilizando o comando `du`, verificar o tamanho do diretório `/etc`, só que exibindo os resultados em *bytes*.

Tarefa 5 – Utilizando o comando `uptime` verificar a quanto tempo o sistema está ativo.

Tarefa 6 – Comando `mount`. O Linux trata suas partições como se fossem diretórios, diferentemente de outros sistemas como *DOS* ou *Windows* que utilizam letras como C: ou A:. No Linux para acessar uma determinada partição de um dispositivo como um CD ou disquete é necessário especificar um ponto de montagem, ou seja, a partir de que diretório este dispositivo será acessado.

O comando `mount` é utilizado para associar um sistema de arquivos a

um diretório. Portanto, para visualizar o conteúdo de um disquete ou CD antes é necessário fazer a montagem.

Colocar um disquete no *drive* de disquete e executar o comando `mount /mnt/floppy`.

Tarefa 7 - Listar o conteúdo do disquete com o comando `ls /mnt/floppy`. Caso o disquete tenha algum conteúdo será exibido, caso esteja vazio, o diretório `/mnt/floppy` também estará vazio.

Tarefa 8 - Utilizando o comando `cp` copiar o arquivo `/etc/passwd` para dentro do disquete. Executar novamente o comando `ls /mnt/floppy`.

Tarefa 9 – O comando `umount` é utilizado para desmontar dispositivos montados pelo comando `mount`. Utilizando o comando `umount` desmontar o dispositivo `/mnt/floppy`.

Tarefa 10 – Utilizando o comando `ls` visualizar o conteúdo do diretório `/mnt/mount`.

Tarefa 11 – Utilizando o comando `mount` montar novamente o disquete e visualizar novamente o conteúdo do diretório `/mnt/floppy`.

Tarefa 12 – Utilizando o comando `mount` e `umount` execute as mesmas operações para o dispositivo `/mnt/cdrom`. Insira no *drive* de CDROOM um CD com algum conteúdo, pode ser o CD de instalação da distribuição em uso.

9.3.4 Criando Apelidos para Comandos (**Alias**)

Criar apelidos para comandos pode facilitar o dia-a-dia. Alguns apelidos, inclusive são utilizados como padrão em algumas distribuições como é o caso do

`ls='ls --color'`, que faz com que toda vez que o comando `ls` é executado seja incluído em seus argumentos o parâmetro “`--color`”. Este parâmetro adicionado ao comando `ls` faz com que a saída do comando diferencie por cores seus diferentes tipos de arquivos (diretórios, arquivos, *links*, executáveis, *etc*).

Os apelidos podem ser criados somente para um usuário como também podem ser criados para todos os usuários do sistema.

Existem duas formas para criar os atalhos, uma é pelo comando executado diretamente na linha de comando `alias` que cria o atalho de forma temporária e a outra é incluindo o comando `alias` nos arquivos de configuração, ou do sistema ou do usuário.

Tarefa 1 – Efetuar *logon* como usuário comum. Utilizando o comando `alias` criar o apelido `cls` para o comando `clear` (`alias cls='clear'`). Testar a execução do `clear` utilizando o apelido `cls`. Digitar no *PROMPT* apenas `cls`.

Tarefa 2 – Efetuar *logon* com outro usuário em outra interface texto e testar a execução do `cls`. Houve a execução do comando como na tarefa 1?

Tarefa 3 – Utilizando o comando `reboot` reiniciar a estação. Após efetuar *logon* com o mesmo usuário da tarefa 1. Testar novamente a execução do `cls`. Houve a execução do comando como na tarefa 1?

Tarefa 4 – Como observado nas tarefas anteriores, a execução do comando `alias` diretamente na linha de comando cria um apelido temporário e somente para o usuário que o executou. Para que o apelido esteja sempre presente para o usuário, o comando `alias` tem que ser incluído no arquivo `.bashrc` que está

no diretório *home* do usuário.

Utilizando o editor de texto Vim, acrescentar a linha indicada em negrito abaixo no arquivo **.bashrc** (`vi .bashrc`). Antes se posicionar no diretório *home* executando o comando `cd`.

```
# User specific aliases and functions  
alias cls='clear'
```

Salvar o arquivo e testar a execução do `cls`. Como observado o comando ainda não funciona. Será necessário fechar o *logon* do usuário utilizando o comando `exit` e efetuar novo *logon*.

Tarefa 5 – Efetuar *logon* com outro usuário em outra interface texto e testar a execução do `cls`. Houve a execução do comando como na tarefa 4?

Tarefa 6 – Utilizando o comando `reboot` reiniciar novamente a estação. Após efetuar *logon* com o mesmo usuário da tarefa 4 e testar o comando `cls`. Observar que mesmo reiniciando a estação o apelido continua funcionando.

Tarefa 7 – O comando `unalias` é utilizado para desfazer um apelido. Logado com o mesmo usuário da tarefa 6, executar o comando `unalias cls`. Executar novamente o `cls` e observar que o apelido não existe mais.

Tarefa 8 – Para que um apelido possa estar presente para todos os usuários do sistema é necessário acrescentar no arquivo `/etc/bashrc`.

- Utilizando o `vi` retirar a linha que foi acrescentada no arquivo `.bashrc` na tarefa 4; (`alias cls='clear'`)

- Efetuar *logon* com o usuário *root* e utilizando o editor Vim acrescentar a mesma linha de comando no arquivo `/etc/bashrc`, como abaixo:

```
# Environment stuff goes in /etc/profile
```

```
alias cls='clear'
```

- Fechar o *logon* do usuário *root* utilizando o comando `exit`;
- Efetuar *logon* com o usuário *root* e executar o comando `cls`. Observar o resultado;
- Efetuar *logon* com outro usuário e executar o comando `cls`. Observar que agora o atalho `cls` está presente para qualquer usuário do sistema.

9.3.5 Agendamento de Tarefas

Os comandos novos utilizados para estas tarefas são: `at` e `crontab`.

Os agendamentos ficam no arquivo `/etc/crontab`. O arquivo `crontab` tem o seguinte formato:

Minuto Hora Dia_do_mês Mês Dia_da_semana Usuário Comando

Exemplo: 30 23 * * * *root* ls /etc > listagem.txt

Tarefa 1 – Pesquisar sobre o `crontab` e responder as questões abaixo:

- Quais os valores válidos para cada campo do arquivo `crontab`:
 - Minuto ->
 - Hora ->
 - Dia do mês ->
 - Mês ->
 - Dia da semana ->

- Qual dia da semana que pode ser representado por dois valores diferentes e quais são estes valores.
- No exemplo dado no início deste item, a listagem do diretório `/etc` será feita em que hora e em quais dias.
- O que é necessário incluir no final do arquivo `crontab` para que todos os comandos sejam executados.

Tarefa 2 – Utilizando o `crontab`, agendar a reinicialização do sistema para que ocorra a 05 minutos do momento atual. Acompanhar a execução do agendamento. Obs.: Fazer o agendamento diretamente no arquivo `/etc/crontab` e logado como usuário `root`.

Tarefa 3 – Pesquisar sobre o comando `at` e agendar a reinicialização do sistema, como foi feito com o `crontab`.

9.3.6 *Shell Script*

Para iniciar esta etapa, será necessário entender primeiro o que é um *Shell* no Linux. *Shell* é um interpretador de comandos, que lê os comandos que são digitados e os converte em forma legível para o sistema operacional. Todo usuário possui um *Shell* atribuído a ele. No Linux existem vários *Shell*, sendo que o mais conhecido e utilizado é o *bash*.

Os *Shell Scripts* são *scripts* com comandos interpretados por um dos *Shell* do Linux. Todos os comandos executados na linha de comando podem ser colocados dentro de um *Shell Script*.

Será visto nesta etapa apenas o básico para criação de um *Shell Script*, pois a intenção aqui é apenas apresentar esta poderosa ferramenta. A referência

indicada para estudo de *Shell Script* é o Livro de Julio César Neves, “Programando *Shell Linux*”, da editora Brasport (NEVES, 2005).

Tarefa 1 – Pesquisar sobre os *Shells* do Linux e citar pelo menos três. Descobrir qual é o *Shell* atribuído ao usuário *root*. Dica: é a última informação do usuário no arquivo `/etc/passwd`.

Tarefa 2 – A primeira informação de um *Shell Script* é qual interpretador será utilizado, após esta informação todos os comandos serão executados segundo aquele interpretador informado exceto as linhas que comecem com “#”. Para entender de forma fácil como isto funciona, os passos abaixo orientam a criação e execução de um pequeno *Shell Script*.

- Se posicionar no diretório *home* utilizando o comando `cd`.

- Utilizando o editor de texto Vim, criar um arquivo com nome de `ola.sh` com o conteúdo abaixo: (`vi ola.sh`).

```
#!/bin/bash
# Meu primeiro Shell Script
clear
echo ""
echo ""
echo "   Ola Mundo !!!!! "
```

- Salvar e sair do arquivo (`:wq!`).

- Para execução de arquivos no Linux é necessário dar permissão para execução.

Utilizando o comando `chmod`, dar ao arquivo `ola.sh` a permissão para execução ao dono. (`chmod 744 ola.sh`).

- Utilizando o comando `ls`, verificar as permissões do arquivo `ola.sh`.

- Executar o *Shell Script*. Observar que se apenas informar na linha de comando o nome do arquivo "`ola.sh`", o Linux retornará a informação de que não encontrou o comando. Neste caso é necessário informar o caminho do diretório em que o arquivo está para que o Linux encontre o arquivo. Executar, portanto desta forma: `./ola.sh`.

Tarefa 3 - Logado como usuário *root*, criar outro *Shell Script* com o nome de `reinicia.sh`, com o conteúdo abaixo:

```
#!/bin/bash  
  
shutdown -r now
```

- Utilizando o comando `chmod` atribuir permissão para execução para o dono do arquivo, no caso o superusuário *root*.

- Executar o *Shell Script* (`./reinicia.sh`) e observar o resultado.

- Após a reinicialização da estação, agendar a execução do *Shell Script* `reinicia.sh` pelo `crontab` para cinco (5) minutos após o momento atual.

Dica: Além do arquivo do *Shell Script* ter as permissões corretas para executar, será necessário informar o caminho completo do diretório que o *script* está armazenado no agendamento que será feito no `crontab`.

9.3.7 Equivalências do *Windows/DOS* e Linux

Abaixo uma relação dos comandos utilizados no *Windows/DOS* e os equivalentes no Linux. Mais informações podem ser encontradas em (SILVA, 2005).

<i>Windows/DOS</i>	Linux	Função
cd diretório	cd diretório	Muda para o diretório indicado
cd..	cd ..	Volta ao diretório anterior
copy	cp	Copia arquivos
del	rm	Remove arquivo
deltree	rm -R	Remove diretórios e subdiretórios
edit	vi, ae, emacs	Editor de texto
format	fdformat, mkfs.ext2	Formata disco
help	man, info	Manual <i>on-line</i>
md	mkdir	Cria diretório
rd	rmdir	Remove diretório
move	mv	Move arquivos
dir	ls	Lista diretórios e arquivos
type	cat	Lista conteúdo de arquivo
print	lpr	Impressão
cls	clear	Limpa a tela
echo	echo	Exibe texto na tela

path	path	Define variáveis de ambiente
type	cat	Concatena arquivos, exibindo o resultado na saída padrão
ver	uname -a	Informações sobre o sistema
date	date	Data e hora
time	date	Data e hora
attrib	chmod	Permissões de pastas e arquivos
scandisk	fsck.ext2	Verifica e repara sistema de arquivos
fdisk	fdisk, cfdisk	Manipula partições do disco
more	more, less	Paginação de arquivos
backup	tar	Utilitário para <i>backup</i>
xcopy	cp -R	Cópia de arquivos

Tarefa 1 – Pesquisar quais são os equivalentes no Linux para os aplicativos relacionados abaixo:

- *Office* –

- *Internet Explorer* –

- *Outlook* –

- *SQL* –

Tarefa 2 – Pesquisar sobre aplicativos Linux para as aplicações abaixo:

- Gravador de *CD/DVD* –
- *Backup* –
- Banco de dados -
- Editoração gráfica –
- Multimídia –
- Navegador –
- Impressão –

10 Apêndice D – Questionários Respondidos

Este Apêndice contém os questionários respondidos pelos quatro voluntários que utilizaram as tarefas para aprendizado do Linux.

10.1 Primeiro voluntário: Isaac Luiz Sarah Sidou

Primeiro Questionário - Perfil do aprendiz:

Nome: **Isaac Luiz Sarah Sidou**

Idade: **46** Sexo: **Masculino**

1 – Qual a sua área de atuação?

Suporte Tecnológico.

2 – Possui algum conhecimento do Sistema Operacional Linux? Se sim, qual?

Sim. Conceitos e noções de comandos.

3 – Possui algum conhecimento de algum outro Sistema Operacional? Se sim qual Sistema Operacional?

Sim. Windows.

4 – Qual é a sua formação? Qual é o seu nível de escolaridade?

Engenharia Elétrica. Superior.

5 – O que motivou o estudo do Sistema Operacional Linux?

Ter uma alternativa de SO para utilização doméstica e na própria empresa

em vista da política de governo de implementação de *software* livre.

6 - Qual será a aplicação prática no seu dia a dia do Sistema Operacional Linux?
No uso doméstico, para desenvolvimento dos conhecimentos do sistema, uso de aplicações baseadas em *software* livre, bem como no trabalho, quando houver a migração de sistemas.

Segundo Questionário - Avaliação do aprendizado utilizando as tarefas.

1 – Relacione abaixo a bibliografia utilizada como complemento as atividades para aprendizado do Linux.

Vários documentos disponíveis na Internet.

2 – Qual distribuição foi utilizada?

Kurumin.

3 – Qual foi a maior dificuldade no aprendizado do Linux?

Utilização inicial dos comandos.

4 – O aprendizado foi individual (autodidata) ou foi feito em grupo (como complemento de curso ou em grupo de estudo)?

Individual.

5 – Quantas horas semanais foram dedicadas ao estudo do Linux?

2 horas.

6 - Quanto tempo foi necessário para completar todas as atividades propostas?

Estudo ainda não foi finalizado.

7 – Em sua opinião, o direcionamento proposto traz um nível de dificuldade crescente no decorrer da apresentação das atividades ou não?

Sim, há um aumento gradativo do grau de dificuldade dos exercícios, promovendo um conhecimento mais avançado dos recursos disponíveis.

8 - Qual sua avaliação da validade da utilização das atividades para o aprendizado do Linux? (O aprendizado seria o mesmo sem a utilização das atividades?).

Criar uma base de aprendizado consistente, em função das atividades propostas.

9 – Qual é a sua avaliação do seu conhecimento em Linux?

Básico, em vista dos imensos recursos existentes no sistema.

10 – Acrescente as observações que achar necessário sobre o uso das atividades para direcionamento do aprendizado do Linux.

10.2 Segundo voluntário: Marcelo Cruz Praude

Primeiro Questionário - Perfil do aprendiz:

Nome: **Marcelo Cruz Praude**

Idade: **24** Sexo: **Masculino**

1 – Qual a sua área de atuação?

Graduando em Psicologia.

2 – Possui algum conhecimento do Sistema Operacional Linux? Se sim, qual?

Não.

3 – Possui algum conhecimento de algum outro Sistema Operacional? Se sim qual Sistema Operacional?

Win98, Win2k e WinXP.

4 – Qual é a sua formação? Qual é o seu nível de escolaridade?

Técnico em Informática.

5 – O que motivou o estudo do Sistema Operacional Linux?

Acompanhar a onda do *software* livre.

6 - Qual será a aplicação prática no seu dia a dia do Sistema Operacional Linux?

Conhecimento alternativo de novas formas de trabalho em computadores pessoais.

Segundo Questionário - Avaliação do aprendizado utilizando as tarefas.

1 – Relacione abaixo a bibliografia utilizada como complemento as atividades para aprendizado do Linux.

Manual Foca GNU/Linux *on-line* e o Manual do Linux (*man*), vários outros documentos disponibilizados na Internet.

2 – Qual distribuição foi utilizada?

Kurumin 5.0

3 – Qual foi a maior dificuldade no aprendizado do Linux?

A maior dificuldade foi em trabalhar na linha de comando.

4 – O aprendizado foi individual (autodidata) ou foi feito em grupo (como complemento de curso ou em grupo de estudo)?

Estudei sozinho.

5 – Quantas horas semanais foram dedicadas ao estudo do Linux?

9 horas.

6 - Quanto tempo foi necessário para completar todas as atividades propostas?

27 horas ao todo.

7 – Em sua opinião, o direcionamento proposto traz um nível de dificuldade crescente no decorrer da apresentação das atividades ou não?

Sim. Passo a passo, a dificuldade é aumentada permitindo uma curva de aprendizagem ascendente e suave.

8 - Qual sua avaliação da validade da utilização das atividades para o

aprendizado do Linux? (O aprendizado seria o mesmo sem a utilização das atividades?).

Creio que a forma proposta de conhecimento com a utilização de atividades é extremamente importante para fixação da tecnologia abordada.

9 – Qual é a sua avaliação do seu conhecimento em Linux?

Suficiente para utilizar uma estação de trabalho no meu dia-a-dia, sem traumas, principalmente com a interface gráfica.

10 – Acrescente as observações que achar necessário sobre o uso das atividades para direcionamento do aprendizado do Linux.

Para o objetivo que foi proposto, as atividades estão bem dimensionadas.

10.3 Terceiro voluntário: Reinaldo Antonio Valério

Primeiro Questionário - Perfil do aprendiz:

Nome: **Reinaldo Antonio Valério**

Idade: **35** Sexo: **Masculino**

1 – Qual a sua área de atuação?

Assistência Técnica em Computadores.

2 – Possui algum conhecimento do Sistema Operacional Linux? Se sim, qual?

Não.

3 – Possui algum conhecimento de algum outro Sistema Operacional? Se sim qual Sistema Operacional?

MSDos, Win3.11, WinNT, Win98, Win2k, WinXP e Novell.

4 – Qual é a sua formação? Qual é o seu nível de escolaridade?

Técnico em Eletrônica.

5 – O que motivou o estudo do Sistema Operacional Linux?

Necessidade Profissional.

6 - Qual será a aplicação prática no seu dia a dia do Sistema Operacional Linux?

Atender as novas necessidades de clientes e usuários.

Segundo Questionário - Avaliação do aprendizado utilizando as tarefas.

1 – Relacione abaixo a bibliografia utilizada como complemento as atividades para o aprendizado do Linux.

Vários materiais encontrados na Internet, principalmente o Guia Foca Linux indicado para estudo, o comando man (manual do Linux).

2 – Qual distribuição foi utilizada?

Fedora.

3 – Qual foi a maior dificuldade no aprendizado do Linux?

Tive maior dificuldade para entender a estrutura de diretórios do Linux.

4 – O aprendizado foi individual (autodidata) ou foi feito em grupo (como complemento de curso ou em grupo de estudo)?

Autodidata.

5 – Quantas horas semanais foram dedicadas ao estudo do Linux?

5 Horas.

6 - Quanto tempo foi necessário para completar todas as atividades propostas?

44 horas ao todo.

7 – Em sua opinião, o direcionamento proposto traz um nível de dificuldade crescente no decorrer da apresentação das atividades ou não?

Sim. Cada vez mais e necessário consultar e leituras adicionais.

8 - Qual sua avaliação da validade da utilização das atividades para o aprendizado do Linux? (O aprendizado seria o mesmo sem a utilização das

atividades?).

É válido, pois ajuda a memorização das funcionalidades dos comandos e ajuda a fixação da sintaxe de cada comando.

9 – Qual é a sua avaliação do seu conhecimento em Linux?

Agregou muito conhecimento que deverei colocar logo em prática.

10 – Acrescente as observações que achar necessário sobre o uso das atividades para direcionamento do aprendizado do Linux.

10.4 Quarto voluntário: Rubens Cruz Praude

Primeiro Questionário - Perfil do aprendiz:

Nome: **Rubens Cruz Praude**

Idade: **20**

Sexo: **Masculino**

1 – Qual a sua área de atuação?

Graduando, 3º ano Engenharia Agrícola.

2 – Possui algum conhecimento do Sistema Operacional Linux? Se sim, qual?

Não

3 – Possui algum conhecimento de algum outro Sistema Operacional? Se sim qual Sistema Operacional?

Win98, Win2k e WinXP.

4 – Qual é a sua formação? Qual é o seu nível de escolaridade?

Graduando, 3º ano Engenharia Agrícola.

5 – O que motivou o estudo do Sistema Operacional Linux?

Pesquisa de Sistemas Operacionais Alternativos

6 - Qual será a aplicação prática no seu dia a dia do Sistema Operacional Linux?

Trabalhar no Futuro com Plataformas Abertas

Segundo Questionário - Avaliação do aprendizado utilizando as tarefas.

1 – Relacione abaixo a bibliografia utilizada como complemento as atividades para o aprendizado do Linux.

Internet, Manual do Linux (man) e Manual Foca.

2 – Qual distribuição foi utilizada?

Kurumin 5.0

3 – Qual foi a maior dificuldade no aprendizado do Linux?

A maior dificuldade foi me habituar a utiliza o modo texto, senti falta do modo gráfico. Mas esta dificuldade foi apenas inicial.

4 – O aprendizado foi individual (autodidata) ou foi feito em grupo (como complemento de curso ou em grupo de estudo)?

Estudei de forma individual.

5 – Quantas horas semanais foram dedicadas ao estudo do Linux?

8 horas.

6 - Quanto tempo foi necessário para completar todas as atividades propostas?

Até agora, umas 25 horas. Ainda não estudei a última parte.

7 – Em sua opinião, o direcionamento proposto traz um nível de dificuldade crescente no decorrer da apresentação das atividades ou não?

Sim, é perceptível um aumento na dificuldade para execução das atividades.

8 - Qual sua avaliação da validade da utilização das atividades para o aprendizado do Linux? (O aprendizado seria o mesmo sem a utilização das

atividades?).

Achei válido sim, visto que existe uma seqüência que agrega conhecimentos a cada tarefa proposta.

9 – Qual é a sua avaliação do seu conhecimento em Linux?

Só o conhecia de ouvir falar.

10 – Acrescente as observações que achar necessário sobre o uso das atividades para direcionamento do aprendizado do Linux.

Satisfiz minha curiosidade.