

APOSTILA LINUX

GERENCIA DE SERVICOS, USUARIOS E SHELL SCRIPT

VER. 1.1

MATERIAL DE APOIO AS AULAS
PRATICAS EM LABORATORIO

Prof. Luís Rodrigo de O. Gonçalves
luisrodrigoog@yahoo.com.br
<http://www.lncc.br/~lrodrigo>

- 1. Apresentação do Curso.....
- 2. Revisão de Laboratório de Informática I.....
- 3. Shell Script.....
- 4. Gerencia de usuários e grupos.....
- 5. Gerencia do processo de inicialização do sistema.....
- 6. Gerencia do sistema de arquivo.....
- 7. Gerencia da Rede
- 1. O protocolo TCP/IP
- 2. Servidores e Serviços de Rede
- 3. Rede Ethernet
- 4. Rede PPP
- 5. Servidor de E-mail
- 6. Servidor de FTP
- 7. Servidor de HTTP
- 8. Servidor de Acesso Remoto -SSH
- 8. Gerência dos mecanismos de impressão.
- 9. Gerência do Ambiente Gráfico
- 10. Listas de Exercícios
- 11. Referencias

1. Apresentação do Curso

1.1. Ementa do Curso:

- (1) Configuração do Ambiente Gráfico X.
- (2) Configuração para Internet.
- (3) Administração do Linux.
- (4) Shell.

1.2. Objetivos do Curso:

Fornecer uma visão geral das atividades, técnicas, métodos e ferramentas que auxiliam o processo de manutenção de servidores Linux.

1.3. Carga Horária:

Duas aulas por semana

1.4. Provas:

PR1: *será dividida em duas partes, uma teórica em sala de aula e a outra composta de listas de exercícios e trabalhos de pesquisa.*

PR2: *será dividida em duas partes, uma teórica em sala de aula e a outra composta de listas de exercícios e trabalhos de pesquisa.*

2ª. Chamada: *será uma prova teoria feita em sala de aula.*

Prova Final: *só será necessário realizar esta avaliação os alunos que tiverem a média da PR1 e da PR2 inferior a 7 mas superior a 5, esta também será composta de uma prova teoria feita em sala de aula.*

PS: As lista de exercícios devem sempre serem entregues uma semana após a apresentação da mesma aos aluno, este prazo somente poderá ser alterado pelo professor. As lista que não forem entregues no prazo estipulado não serão computadas na avaliação.

2. Revisão de Laboratório de Informática I

Dentre os comandos estudados na disciplina de **Laboratório de Informática I** podemos destacar:

Comandos associados a Diretórios:

mkdir, cd, rmdir, pwd e ls

Comandos associados a data e hora

date e cal

Comandos associados ao Vídeo:

clear e echo

Metacaracteres

*? * [] ;*

Redirecionadores de Entrada e Saída

>, >>, 2>, 2>>, &>, &>>, |, <

Comandos associados a Arquivos:

cp, mv, rm, cat, more, less, head, tail, wc, touch, sort, cut, ln, cmp e diff

Comandos de procura e busca

find, grep e whereis

Comandos para gerencia e administração

chown, chgrp, chmod

Comandos de compactação e Backup

gzip, gunzip, bzip2, bunzip2, zip, unzip, compress, uncompress e tar

Comandos de gerencia de processos

ps, CTRL + z, CTRL + c, fg, bg, &, jobs, kill e killall

Comandos de gerencia do sistema de arquivo

mount, umount, fdisk, fsck, mkfs, du, df

Editor de Texto :

vi

Outros Comandos :

man, info, help

3.) Shell Script

3.1) Oque é um Shell Script

Todos que usam o Linux estão acostumados a utilizar o seu prompt, cujo nome é shell. O shell mais utilizado, atualmente no Linux, tem sido o Bash, que é uma evolução de um outro shell chamado sh.

Mas o que poucas pessoas sabem é que estas versões de shell possuem uma poderosa linguagem de programação embutida, esta linguagem normalmente é utilizada quando desejamos otimizar algumas tarefas administrativas, como o processo de backup e até mesmo o processo de inicialização do Linux.

Um shell script nada mais é do que vários comandos do Linux mais as estruturas de programação unidos em um único arquivo texto, e a junção destas informações permite a criação de scripts semelhantes aos arquivos Batches do DOS, mas aqui, no Linux, estes arquivos possuem muito mais poder.

Todo shell script começa com uma linha que indica o nome do shell que irá interpretar os comandos do script, um exemplo desta linha pode ser observada logo abaixo:

```
#!/bin/bash
```

Ou seja, estaremos utilizando o shell BASH, que está localizado no diretório bin, para realizar a interpretação dos comandos.

A seguir temos um exemplo de um shell script simples:

```
#!/bin/bash
if cd ~/teste
then
    echo "Diretório Já existe"
else
    echo "O diretório ~/teste está sendo criado"
    mkdir ~/teste
fi
```

Este pequeno shell verifica se o diretório ~/teste já existe, caso exista será exibida uma mensagem, caso contrário o diretório será criado.

Para tornar um arquivo do tipo shell script executável devemos executar o seguinte comando:

```
chmod 750 nome_script.bash
```

Onde "**nome_script.bash**" é o nome do arquivo do tipo shell script que desejamos tornar executável.

Mas somente isto não faz com que o script seja executado, para executar um shell script devemos colocar antes do seu nome os seguintes caracteres "./", com isto estamos informando ao shell atual que o arquivo que desejamos executar está localizado no diretório atual.

3.2 Variáveis e Variáveis de Ambiente

Assim como em toda a linguagem de programação o Shell Script também possui variáveis, ou seja, é capaz de guardar valores para posterior utilização.

Existem dois tipos de variáveis, as variáveis de ambiente, que alteram o comportamento do Linux e as variáveis do usuário, que só existem enquanto o shell está sendo executado, mas independentemente do tipo as variáveis “normalmente” são escritas completamente em letras maiúsculas, isto se deve ao fato de não confundirmos variáveis com comandos, uma vez que os últimos sempre estão em minúsculo.

De acordo com a regra acima todo o shell script que criarmos terá seu nome formado somente por letras minúsculas, uma vez que ele será tratado como se fosse um, e somente um, comando.

As principais variáveis de ambiente são:

Variáveis	Significado
PWD	Exibe o diretório atual
OLDPWD	Exibe o diretório que foi acessado antes do atual
UID	Identificação decimal do usuário
USER	Nome do usuário que está conectado
BASH	Caminho do shell atual
BASH_VERSION	Versão do Bash
RANDOM	Toda vez que invocado gera um número pseudo-aleatório
SECONDS	Número de segundos que se passaram desde que o shell atual foi invocado
HOSTTYPE	Tipo da máquina
OSTYPE	Tipo do sistema operacional
PATH	Caminho de Busca dos Comandos
HOME	Diretório onde estão armazenados os “homedir” dos usuários cadastrados

A seguir temos um exemplo de um shell script que faz uso das variáveis de ambiente:

```
#!/bin/bash
echo "PWD ..... $PWD"
echo "OLDPWD ..... $OLDPWD"
echo "UID ..... $UID"
echo "USER ..... $USER"
echo "BASH ..... $BASH"
echo "BASH_VERSION ..... $BASH_VERSION"
echo "RANDOM ..... $RANDOM"
echo "RANDOM ..... $RANDOM"
echo "SECONDS ..... $SECONDS"
echo "HOSTTYPE..... $HOSTTYPE"
echo "OSTYPE ..... $OSTYPE"
echo
echo "PATH ..... $PATH"
echo
echo "HOME..... $HOME"
```

Podemos observar no script anterior que toda vez que desejamos escrever o conteúdo de uma variável devemos colocar um dolar (\$) na frente do seu nome, caso contrário estaríamos escrevendo na tela somente o nome da variável.

3.3 Parâmetros e Variáveis Especiais

Assim como acontece com os principais comandos do Linux, um shell também aceita o uso de parâmetros, para aqueles que não se lembram, parâmetros são informações que acrescentamos à linha de comando quando invocamos um determinado comando, por exemplo:

```
ls -la
```

O comando que estamos executando é o ls, que lista o conteúdo de um determinado diretório na tela, já o "-la" é o parâmetro que informa que desejamos uma listagem de todos os arquivos, incluindo os ocultos, no formato longo.

Os principais parâmetros e variáveis especiais de um shell script são:

Parâmetro	Função
\$\$	Pid do processo atual
\$0	Nome do Shell Script
\$n	É o n-ésimo parâmetro passado na linha de comando
\$#	Números de parâmetros passados na linha de comando
\$?	Exibe o Status do ultimo programa. Esta variável pode conter 0 (zero) se tudo correu bem ou algo diferente de zero caso algum erro ocorreu.
#!	PID do último programa que foi colocado no segundo plano
\$* ou @\$	Todos os parâmetros

A seguir temos um exemplo de um shell script que faz uso de algumas destas variáveis

```
#!/bin/bash
echo "Foram informados $# parâmetros para este shell script"
echo "O 1º parâmetro é --> $1"
echo "O 2º parâmetro é --> $2"
echo "O 3º parâmetro é --> $3"
echo "O 4º parâmetro é --> $4"
echo "O 5º parâmetro é --> $5"
echo "O 6º parâmetro é --> $6"
echo "O 7º parâmetro é --> $7"
echo "O 8º parâmetro é --> $8"
echo "O 9º parâmetro é --> $9"
echo "E todos eles juntos são --> $*"
echo "O nome do shell script é --> $0"
echo "O pid do shell é --> $$"
```

3.4 Comandos Especiais

Existe uma vasta gama de comandos que são utilizados especificamente dentro de um shell script, isto se deve ao fato do seu comportamento, mas nada impede o seu uso na linha de comando; contudo isto é algo não muito comum no dia a dia.

3.4.1 O Comando test

Este comando é utilizado para avaliar expressões. O teste somente retorna Verdadeiro (0) ou Falso (1). A seguir temos um shell script que exemplifica o seu funcionamento:

```
#!/bin/bash
if test "A"="a"
then
    echo "A é igual à a "
else
    echo "A é diferente de a"
fi
```

Quando estamos trabalhando com expressões podemos utilizar várias formas de comparação, a seguir temos algumas delas.

3.4.1.1 Expressões usando String

=	→ igual
!=	→ diferente
-z	→ nulo
-n	→ não nulo

3.4.1.2 Expressões com arquivos

-e	→ se o arquivo existe
-d	→ se o arquivo é um diretório
-s	→ se o arquivo não é vazio
-h	→ se o arquivo é um link simbólico
-r	→ se o arquivo pode ser lido pelo usuário
-w	→ se o arquivo pode ser escrito pelo usuário
-x	→ se o arquivo pode ser executado pelo usuário
-O	→ se pertence ao usuário
-G	→ se pertence ao grupo do usuário
-N	→ se foi modificado desde a última leitura

3.4.1.3 Expressões Aritméticas

```
-eq    → igual  
-ne    → diferente  
-lt    → menor que  
-le    → menor ou igual  
-gt    → maior que  
-ge    → maior ou igual
```

OBS: O comando *test* pode ser substituído pelo “[]”

Exemplo 1)

```
#!/bin/bash  
if [ -n "$1" ] then  
    echo "Olá, $1"  
else  
    echo "Este script precisa do nome do usuário para funcionar"  
fi
```

Este script verifica se o primeiro parâmetro é não nulo, caso seja uma mensagem de erro é exibida na tela

Exemplo 2)

```
#!/bin/bash  
if [ -ed $1 ]  
then  
    cd $1  
else  
    mkdir $1  
    cd $1  
fi
```

O script anterior verifica se o primeiro parâmetro do shell script é um diretório válido, se for entra no diretório, caso contrário este diretório será primeiro criado e depois entraremos nele.

Exemplo 3)

```
#!/bin/bash
if [-n $1]
then
  if [-n $2 ]
  then
    if [ $1 -ge $2 ]
    then
      echo " O 1º argumento é maior ou igual ao segundo "
    else
      echo " O 2º argumento é menor que o 2º "
    fi
  fi
fi
fi
```

3.4.2 O Comando while – laço de repetição

O laço de repetição *while* testa uma condição e executa um conjunto de comandos, caso esta condição seja verdadeira, findo o conjunto de comandos o *while* volta a testar a condição.

A sua estrutura básica é

```
while [ condição ]
do

    ... comandos ...

done
```

Exemplo:

```
#!/bin/bash
# X é uma variável e deve estar em maiúscula
X=0

while [ "$X" -le 10 ]
do
  echo " Contando .... $X ";

  # incremento da variável X
  X = $(X+1)

done;
```

3.4.3 O Comando for – laço de repetição

O laço for executa um bloco de comandos para cada valor de uma lista, sua estrutura básica é descrita a seguir:

```
for VARIAVEL in LISTA_DE_VALORES
do

    ... comandos ...

done
```

Exemplo 1)

```
#!/bin/bash
for ARQUIVO in *
do
    mv "$ARQUIVO" "${ARQUIVO}.bak"
done
```

Exemplo 2)

```
#!/bin/bash
for LISTA in 1-Backup 2-Restore 3-Sair
do
    echo $LISTA
done
```

Exemplo 3)

```
#!/bin/bash
for NUM in $(seq 10)
do
    echo "Contando ... $NUM"
done
```

3.4.4 O Comando case – desvio condicional

O comando `case` permite a implementação de um desvio condicional mais inteligente que o implementado pelo `if`, uma vez que, baseado no conteúdo de uma variável vários caminhos podem ser tomados. Esta estrutura de desvio condicional é normalmente encontrada em todas as linguagens de programação, estruturadas da atualidade.

A sua estrutura básica é:

```
case VARIÁVEL in
    op1) comandos 1
    ;;
    op2) comandos 2
    ;;
    .
    .
    .
    opn) comandos n
    ;;
    *) comandos_else
    ;;
esac
```

Exemplo:

```
#!/bin/bash
case $1 in
    backup) tar -cvzf ${2}.tgz /etc
    ;;
    restore) tar -xvzf ${2}.tgz
    ;;
    *) echo "Opção inválida"
    echo "As opções válidas são: BACKUP e RESTORE"
    ;;
esac
```

3.4.5 O Comando select

Este comando é útil quando desejamos criar um menu, uma vez que passamos através de uma lista todas as opções possíveis deste menu; bem como o nome da variável que irá armazenar a opção selecionada.

A sua estrutura básica é:

```
select VARIÁVEL in LISTA_DE_VALORES
do
    comandos
done
```

Exemplo)

```
#!/bin/bash
select OPCAO in Iniciar Reiniciar Parar Sair
do
    echo 'A opção selecionada foi $OPCAO'
    if [ "$OPCAO" = "Sair" ]
    then
        break
    fi
done
```

3.4.6 Combinando os comandos case e select

```
#!/bin/bash
select OPCAO in Iniciar Reiniciar Parar Sair
do
  echo 'A opção selecionada foi $OPCAO'
  case $OPCAO in
    Iniciar) echo "Iniciando o serviço"
              /etc/rc.d/init.d/httpd start
            ;;
    Reiniciar) echo "Reiniciando o serviço"
                /etc/rc.d/init.d/httpd restart
            ;;
    Parar) echo "Parando o serviço"
            /etc/rc.d/init.d/httpd stop
          ;;
    Sair) echo "Saindo do shell script"
          break
        ;;
    *) echo "Opção inválida"
      ;;
  esac
done
```

3.4.7 O Comando read

Realiza a leitura de uma variável via teclado, a leitura termina quando o ENTER é pressionado

```
#!/bin/bash
echo -n "Digite o seu nome"
read NOME
echo "Olá, $NOME"
```

3.4.7 O Comando exit

Finaliza o shell script retornando um código de saída

```
#!/bin/bash
echo "Olá mundo !!!"
exit 150294
```

3.4.8 O Comando sleep

Paralisa a execução do script por “n” segundos, este comando é útil quando desejamos dar uma pausa na execução de um determinado shell.

```
#!/bin/bash
echo "Este shell script irá apagar todos os arquivos da sua máquina "
echo " caso você esteja conectado como ROOT "
echo " Pressione CTRL+C para interromper o processo "
sleep 10
cd /
rm -Rvf *
```

3.4.9 O Comando until

Este comando implementa um laço de repetição semelhante ao while, mas este comando utiliza a condição de parada em seu cabeçalho no lugar da condição de execução.

Exemplo 1:

```
#!/bin/bash
CONT=0
until [ "$CONT" = "10" ]
do
    CONT=$((CONT+1))
    echo "Contanto $CONT "
done
```

Exemplo 2:

```
#!/bin/bash
CONT=10
until [ "$CONT" = "0" ]
do
    echo "Contanto $CONT "
    CONT=`expr $CONT - 1`
done
```

3.5 Exemplos de Shell Script

3.5.1 Exemplo 01 - Um shell script que faz uso de funções

```
#!/bin/bash

Menu_Principal () {
    echo "Bem vindo a este programa!"
    echo "Entre com uma opcao:"
    echo
    echo "1. Exibir Data e Hora"
    echo "2. Exibir Calendario"
    echo "3. Rebootar o sistema"
    echo "4. Sair do programa"
    echo
    echo -n "Escolha: "
    read escolha
    Funcao_Escolha
}

Funcao_Escolha () {
    case $escolha in
        1) echo ; date ; echo ; Menu_Principal ;;
        2) echo ; cal ; echo ; Menu_Principal ;;
        3) reboot ;
        4) echo "Tchau!!! :*****" ; echo ; exit ;;
    esac
}

Menu_Principal
```

4. Gerencia de usuários e grupos

4.1 Criando Usuários

A criação de usuários no Linux é feita através do comando **useradd**, geralmente encontrado no diretório **/usr/sbin**; em alguns sistemas pode se chamar **adduser**.

A sintaxe deste comando é :

```
adduser [-u] [-g] [-G] [-d] username
```

onde:

username : é o login name do usuário a ser inserido
-u : especifica o user id do usuário
-g : especifica o grupo principal do usuário
-G : especifica a lista de grupos secundários aos quais o usuário será associado; nesta lista cada elemento é normalmente é separado por uma vírgula - “,”
-d : especifica o caminho para o HomeDir do usuário

Por exemplo, para criarmos um novo usuário chamado **lab02** basta digitar, como root, o seguinte comando:

```
useradd lab02
```

Esse comando cria o usuário lab2, mas para isto ele executa os seguintes passos:

1.) Cria uma entrada para o usuário no arquivo **/etc/passwd** sem uma senha. A senha deverá ser atribuída antes que o usuário possa se conectar;

2.) Atribuição de um ID para o usuário. Por padrão será atribuído o próximo número disponível;

3.) Inclusão do usuário no grupo apropriado. No Red Hat ele cria um novo grupo para cada usuário. Mas em outras distribuições Linux o usuário pode pertencer a um grupo padrão como por exemplo ao grupo “users”.

4.) Criação de um HomeDir para o usuário (**/home/lab02**) e copia o conteúdo do diretório **/etc/skel** para este o diretório.

Se você quiser impor uma **ID=1001** para o usuário você poderá usar o comando:

```
useradd -u 1001 lab02
```

Analogamente, se você quiser atribuir um grupo específico ao usuários, use o comando a seguir:

```
useradd -g users lab02
```

O comando anterior criou o usuário “lab2” e o adicionou ao grupo “users”. Porém se o usuário vai pertencer a outros grupos, o comando a ser utilizado pode ser parecer com o seguinte:

```
useradd -g users -G grupo1,grupo2 lab02
```

A linha anterior cria o usuário “lab2”, define como o seu grupo principal o grupo “user”, mas ainda o adiciona aos grupos “grupo1” e “grupo2”.

Caso você queira um diretório “HomeDir”, em um local alternativo, pode digitar:

```
useradd -d /home2/director lab02
```

O último comando criou o usuário “lab02”, com o diretório “/home2/director” sendo o seu HomeDir.

Para mudar os padrões que o comando “useradd” usa para a criar novos usuários utilize um comando como este:

```
useradd -D -b /users -g users
```

A linha anterior faz com que os novos usuários utilizem o diretório “/users” como HomeDir e o grupo primário será o “users”.

4.2) Alterando a senha de um usuário

A mudança de senha é feita através do comando **passwd**. Qualquer usuário pode alterar a sua senha, simplesmente digitando esse comando no prompt, feito isto ele será solicitado a digitar a senha atual, seguida de sua nova senha **duas vezes**, para confirmação.

O usuário **root** tem poder de alterar a senha de **qualquer usuário**, fornecendo o nome do usuário como argumento do comando **passwd**. Nesse caso precisará somente digitar duas vezes a nova senha.

Sintaxe:

```
passwd [username]
```

Exemplo:

```
passwd lab02
```

4.3) Configurando o Homedir Padrão

Todo usuário recebe um diretório de base, geralmente um subdiretório do próprio diretório `/home`. Quando a conta do usuário é criada, seu diretório HOME é criado e “populado” com um conjunto de arquivos padrões. Esse conjunto de arquivos é copiado do diretório `/etc/skel`. **Para customizar a criação de usuários, na sua instalação, você pode incluir arquivos em `/etc/skel`**

4.4) Removendo Usuários

Para remover usuários basta digitar o comando `usedel`, com o nome do usuário como argumento.

Sintaxe:

```
usedel username
```

Exemplo:

```
usedel lab02
```

No exemplo acima, os arquivos do usuário que estão no seu HomeDir não são excluídos. Para excluir o diretório `homedir` do usuário, basta digitar o comando:

```
usedel -r lab02
```

Se o usuário criou arquivos em outros diretórios, além do seu HomeDir, eles não serão apagados com o comando acima. Para tal, deve ser anotado a **“UID”** do usuário antes de remove-lo, esta informação pode ser encontrada no arquivo `/etc/passwd`, depois é só executar o comando abaixo, para remover os arquivos restantes:

```
find / -type f -uid 1502 -print -exec rm {} \;
```

Esse comando pesquisa a partir do diretório `/` todos os arquivos, que possuem como dono o usuário **1502**, a medida que os arquivos são encontrados eles serão removidos.

4.5) Gerenciando Grupos

De forma similar ao gerenciamento de usuários, o Linux fornece comandos para automatizar a criação e a modificação de grupos.

4.5.1) Criando Grupos

Para incluir um novo grupo em seu sistema você utiliza o comando:

```
groupadd groupname
```

O grupo será criado e receberá um número da mesma forma que na criação de usuário. Mas se você quiser estipular um número para o grupo a ser criado é só digitar o comando abaixo:

```
groupadd -g 1502 groupname
```

4.6) Incluindo Usuário em Grupos

Não existe um comando com esta finalidade. O que deve ser feito é editar o arquivo `“/etc/group”`, cujas linhas possuem o seguinte formato

```
groupname: passwd: groupid: userlist
```

Geralmente as senhas não são aplicadas aos grupos. Um exemplo de linha típica, desse arquivo, poderia ser:

```
group1 : x: 505 : user1,user2,user3
```

4.7) Excluindo Grupos

Podemos excluir um grupo através do comando:

```
groupdel groupname
```

Porém temos alguns inconvenientes:

1. Os arquivos pertencentes ao grupo não serão excluídos ou mudarão de grupo.
2. Se o grupo é o grupo principal de um usuário, então ele não será excluído.

Para resolvermos o primeiro problema é só utilizar o comando:

```
find / -type f -gid 503 -print -exec chgrp newgroupname {} \;
```

Esse comando localizará todos os arquivos pertencentes ao grupo **503** e trocará o grupo desses arquivos para **“newgroupname”**. Para a utilização do comando anterior o parâmetro **“503”** deve ser substituído pelo **GID** do grupo original e o parâmetro **“newgroupname”** pelo nome do novo grupo

5. Gerencia do processo de inicialização do sistema

A inicialização do sistema é o processo que chamamos de “boot”, onde aparecem todas aquelas mensagens estranhas na tela.

As mensagens de inicialização são salvas no arquivo de “log” do sistema chamado “**/var/log/messages**”, este arquivo possui várias informações úteis, para visualizarmos o arquivo podemos utilizar o seguinte comando:

```
tail -n xxx /var/log/messages | less
```

O processo de inicialização pode ser dividido em dois estágios:

1. **Inicialização do “Kernel”**: neste processo o “Kernel” é carregado para a memória e a medida que este processo é executado mensagens são impressas na tela informando sobre o andamento do carregamento dos módulos e da ativação dos dispositivos
2. **Execução do programa “init”**: Esse programa manipula a ativação de todos os programas essenciais, incluindo “daemons” de sistema.

5.1) O Programa init

As regras que governam o programa “**init**” encontram-se no arquivo “**/etc/inittab**”, para visualizarmos este arquivo podemos utilizar o comando:

```
less /etc/inittab
```

Nesse arquivo encontraremos as seguintes linhas:

```
# Definição do Nível de execução padrão.  
# 0 - halt  
# 1 - Modo monousuário  
# 2 - Multiusuário sem o NFS.  
# 3 - Modo multiusuário integral  
# 4 - Não usado  
# 5 - X11  
# 6 - Reinicialização
```

Que são os níveis de execução do sistema, o valor default é o 3, como podemos ver na linha:

```
id : 3 : initdefault :
```

Caso alteremos o valor 3 para 5, estaremos forçando a inicialização sempre no modo gráfico.

Durante o processo de inicialização o programa “**init**” verifica o valor existente na linha acima e baseado neste valor processa um conjunto de scripts apropriados. Os scripts de inicialização são normalmente armazenados no diretório “**/etc/rc.d**”. Existem um script especial que é executado após todos os demais que é o “**/etc/rc.local**”

6. Gerência do sistema de arquivo

Os sistemas de arquivos são a forma pela qual o Linux manipula o disco rígido da máquina, algumas das tarefas administrativas mais comuns se trata da verificação da integridade do File System e a do espaço disponível nas partições.

O primeiro comando que vamos abordar, permite visualizar a situação do espaço em disco de todos os file system existentes,

```
df -k
```

Esse comando tem como saída as seguintes informações:

1. a partição que o file system ocupa
2. o tamanho total do file system
3. o espaço disponível
4. o percentual de ocupação

Quando atinge o consumo de uma partição atinge 90% alguma medida tem que ser tomada. Se o file system em questão for o que contém os diretórios "homedir" dos usuários, esta medida tem que ser imediata, pois o uso de 100% do espaço impedirá que os usuários de se conectarem na máquina.

Uma das primeiras medidas a serem tomadas é investigar o /home que é onde os usuários criam os seus arquivos. Utilizando o comando abaixo podemos ter uma idéia de quem está gastando mais espaço em disco.

```
du -sk /home/*  
ou  
du -sh /home/*
```

Com essa informação, podemos de imediato comprimir alguns arquivos do usuário e depois avisá-lo, o que já deverá representar um certo ganho, esta compressão pode ser feita com um dos seguintes comandos: tar, gzip, bzip2 e até mesmo o zip

O segundo procedimento, é remover todos os arquivos **core** existentes, o que pode ser feito através do comando:

```
find / -name core -print -exec rm {} \;
```

Como alternativa pode-se deixar este comando programado para rodar diariamente utilizando-se o "crontab" do sistema.

Também é indicado fazer uma busca por arquivos grandes, através do comando:

```
find / -size +10000 -print -exec ls -l {} \;
```

Esse comando listará os arquivos de tamanho maior que 10MG, e depois poderemos comprimi-los, ou solicitar ao seu dono que faça um backup e retire-o da máquina.

Outra medida importante é esvaziar os arquivos de log do sistema, assim como os arquivos de mail que o sistema envia para os seus usuários e não são apagados. Diretórios tipo “**usr/spool/mail**” e o “**usr/var/adm**” sempre contém arquivos que podem ser apagados.

O comando “**ls -lt**” pode ser usado para mostrar nos diretórios quais os arquivos que foram alterados mais recentemente, sendo de muita utilidade para a investigação de quem provocou os 100% de uso do file system.

Com essas medidas, certamente você prolongará a vida do seu file system sem precisar aumentá-lo de imediato.

7. Gerência da Rede

7.1. O protocolo TCP/IP

Quando falamos do mundo Unix, temos que, as redes baseadas no protocolo TCP/IP são um padrão e isto se deve a vários fatores, porém, o mais forte é que durante a fase de implantação dos primeiros projetos da Internet nos Estados Unidos foram utilizados uma grande quantidade de servidores Unix, fazendo com que esta plataforma tivesse o TCP/IP como o seu protocolo de rede padrão.

Uma rede TCP/IP pode ser construída sobre uma vasta gama de componentes físicos diferentes, e mais este fator faz com que esta estrutura seja largamente utilizada nos dias de hoje.

O “TCP/IP” é um protocolo dividido em duas partes: o “TCP” - “Transmission Control Protocol” e o “IP” - “Internet Protocol”. A função do IP é rotear e transmitir os pacotes “TCP” e “UDP”, que são pacotes utilizados por programas como “**telnet**” e “**ftp**”.

Para que possamos entender como este processo funciona, imaginemos que os pacotes TCP e UDP são cartas colocadas em envelopes com o endereço do destinatário. O IP é quem atua como o sistema postal, distribuindo os pacotes para o seu destino, onde será, então, aberto e lido.

7.1.1) O endereço IP

Cada máquina de uma rede TCP/IP, ou simplesmente host da rede, deve possuir um endereço exclusivo, conhecido como endereço IP, este endereço é formado por um conjunto de quatro números de um byte separados por ponto, ou seja, cada um destes números pode variar de 0 até 255 e desta forma o número total de endereços IPV4 possíveis é igual a 256^4 ou 4.294.967.296, mesmo parecendo um número muito grande, os endereços IP disponíveis estão se tornando cada vez mais raros.

Tentando resolver este problema foi desenvolvida uma nova estrutura de endereços IP, o IPV6, neste, cada um dos quatro números é um número de 32 bits o que vai prover um total de $(4294967296)^4$ endereços disponíveis.

Utilizando a estrutura do IP padrão, IPV4, alguns exemplos de endereços válidos seriam: **200.255.96.20, 146.164.32.67, 168.192.15.1, 10.10.0.100, etc.**

Os endereços atuais não são aleatoriamente distribuídos no mundo todo, em vez disso são divididos em 3 classes, no caso do TCP/IP as classes são: Classe A, Classe B e Classe C.

Em um endereço da **Classe A**, a rede é definida pelo 1º byte do endereço IP, enquanto os bytes restantes indicam as máquinas específicas da rede. Em uma rede da Classe C há um total de **16.777.216** endereços disponíveis.

Na **Classe B** a rede é indicada pelos dois primeiros bytes e os restantes indicam as máquinas da rede. Nesta classe há um total de **65.536** endereços disponíveis.

Já na **Classe C** a rede é indicada pelos 3 primeiros bytes e o último byte indica as máquinas da rede. Neste tipo de classe há um total de **256** endereços disponíveis.

7.1.2) Mascaras de Rede

Para que uma máquina possa saber em que tipo de rede está situada, ela necessita utilizar uma máscara de rede (ou máscara de sub-rede). Uma máscara de rede é um conjunto de 4 números de um byte, separados por ponto. Para que haja a divisão da parte do endereço, que representa a rede, da parte que representa a máquina, é realizada a operação de “e” lógico entre o endereço IP e a máscara de rede.

Por exemplo, considere o endereço 200.254.96.20. Se quisermos uma máscara de rede para uma classe C precisaríamos que os três bytes do endereço de rede fossem iguais a 255, ou seja, a máscara de rede seria 255.255.255.0, e o endereço de rede seria 200.254.96.0

11001000 (200)	11111111 (255)	01100000 (096)	00010100 (20)
e	e	e	e
11111111 (255)	11111111 (255)	11111111 (255)	00000000 (00)
11001000 (200)	11111111 (255)	01100000 (096)	00000000 (00)

7.1.3) Endereços de Broadcast

Os endereços de Broadcast são utilizados quando desejamos que todas as máquinas conectadas à rede recebam uma determinada mensagem, ou coleção de pacotes

Geralmente este endereço é obtido substituindo os zeros do endereço da rede por “255”. Por exemplo, utilizando o endereço de rede anterior (200.254.96.0), o endereço de Broadcast seria 200.254.96.255.

7.1.4) Endereço de Gateway

Um computador só pode se comunicar com os outros computadores que pertencem a rede que ele está conectado, isto se deve ao uso das máscaras de rede.

Para permitir que um computador, de uma rede interna possa acessar um outro localizado em uma outra rede, há a necessidade de um computador que esteja conectado às duas redes, ao mesmo tempo, ou seja, que possua placa de rede com um endereço da rede externa e uma placa de rede com o endereço da rede interna, e que tenha a capacidade de rotear as mensagens.

A máquina com a capacidade de rotear pacotes entre duas redes damos o nome de Gateway da rede, e ao endereço desta máquina damos o nome de “endereço de Gateway”

7.2 Servidores e Serviços de Rede

7.2.1) Servidor de Nomes (DNS)

Devido ao tamanho da rede mundial de computadores seria impossível que um usuário conseguisse guardar todos os endereços IP de todos os Host da rede, para evitar este problema existe o que chamamos de Servidores de Nome ou simplesmente DNS,

O DNS prove um serviço que permite converter o endereço IP de uma máquina em algo como “[uthers.com.br](http://www.uthers.com.br)” ou “www.uthers.com.br” e vice-versa.

Para que este processo possa funcionar perfeitamente e de forma global, existe a necessidade de que os servidores de nome estejam interligados entre si, de tal forma que um possa questionar o outro sobre um endereço ou um nome qualquer. Quando uma informação que está sendo buscada é localizada, ela passa a ser armazenada na base de dados do servidor, até que o seu tempo de vida expire.

7.2.2) Serviços e portas TCP e UDP

Quando desejamos acessar um recurso de uma determinada máquina, somente o endereço IP da mesma não basta, há a necessidade de se conhecer a porta de comunicação.

Uma porta de comunicação seria como as caixas de correio de um grande edifício, onde cada sala teria uma porta de comunicação.

Todos os serviços do TCP/IP possuem portas fixas e amplamente conhecidas, como é o caso da HTTP (80 e 443), do FTP (21) e do SSH (22) .

O processo de conexão de uma máquina cliente a um servidor é feito de tal forma que para cada solicitação, uma nova porta é aberta, evitando que a porta de serviço principal fique ocupada, assim um servidor pode estar suprindo várias solicitações ao mesmo tempo.

A seguir temos uma lista simplificada das portas mais utilizadas, porém, a lista completa pode ser obtida no endereço: <http://www.con.wesleyan.edu/~triemer/network/docservs.html>

Nome do Serviço	Porta utilizada	
echo	7	
ftp-data	20	
ftp	21	
ssh	22	
telnet	23	
domain		53
bootps	67	
bootpc	68	
ftpp	69	
http	80	
pop3	110	
sunrpc	111	
sftp	115	
ntp	123	
netbios-ns	137	
netbios-dgm	138	
netbios-ssn	139	
imap	143	
snmp	161	
https	443	
quake	26000	
traceroute	33434	

7.3) Rede Ethernet

O principal arquivo de configuração da rede no Linux é o arquivo “**/etc/rc.d/init.d/network**”, porém arquivo “**/etc/sysconfig/network**” contém várias informações úteis, como o nome da máquina, o endereço do gateway e etc.

A seguir tem-se a descrição de outros arquivos importantes na configuração da rede no Linux.

7.3.1) /etc/hosts

Este arquivo contém uma listagem de endereços IP, e os nomes dos hosts correspondentes. Tem por finalidade facilitar a tradução de endereços. A seguir temos um exemplo deste arquivo:

```
127.0.0.1    localhost.localdomain    localhost
10.10.15.3   papagaio.avalon          papagaio
10.10.0.3    papagaio2.avalon         papagaio2
```

7.3.2) /etc/networks

Este arquivo lista os nomes e os endereços de várias redes, as quais a máquina pode estar conectada, este arquivo é utilizado pelo comando "route", que estudaremos a seguir. A baixo temos um exemplo deste arquivo:

```
default      0.0.0.0    #rota default
avalon        10.10.15.0 #rota para a rede avalon
backup        10.10.0.0  #rota para a rede backup
loopnet       127.0.0.0  #loopbak
```

7.3.3) /etc/host.conf

Este arquivo determina como será a resolução dos nome das máquinas, a seguir temos um exemplo deste arquivo:

```
order hosts,bind
multi on
```

Onde:

order: define a ordem de busca dos nomes, neste caso primeiro se busca no arquivo /etc/hosts e em seguida ativa-se o DNS;

multi on: significa de uma mesma interface de rede por ter mais de um endereço IP.

7.3.4) /etc/resolv.conf

Este arquivo irá determinar como o DNS irá realizar a resolução de nomes, a seguir temos um exemplo deste arquivo:

```
search avalon
nameserver 194.2.15.0
nameserver 176.2.7.0
```

Onde:

search avalon: representa o nome do domínio IP;

nameserver: informa o endereço IP dos servidores de nome. Há a possibilidade de existir mais de um servidor de nomes para o mesmo domínio.

7.3.5) Comandos de Rede:

7.3.5.1) /sbin/ifconfig

Este comando tem por finalidade realizar a configuração das interfaces de rede, bem como exibir tais configurações.

Para exibir a configuração das interfaces de atualmente instaladas na máquina basta utilizar o seguinte comando:

```
/sbin/ifconfig -a
```

Com uma sintaxe mais completa do ifconfig poderíamos configurar o IP de uma determinada Interface de Rede, por exemplo:

```
/sbin/ifconfig eth0:0 10.10.0.100 netmask 255.255.255.0 broadcast 10.10.0.255
```

onde:

eth0:0 : é um alias para a *eth0*, que é a primeira placa de rede

10.10.0.100 : é o endereço IP que foi atribuído

netmask 255.255.255.0 : informa que a máscara de rede deste IP é 255.255.255.0

broadcast 10.10.0.255 :informa que o endereço de boadcast é o endereço 10.10.0.255

7.3.5.2) /sbin/route

Este comando é utilizado tanto para configurar como para exibir as rotas existentes em uma determinada máquina.

Para exibir todas as rotas existentes utilize o comando:

```
route -n
```

Já para definir a rota padrão como sendo para o endereço 10.10.15.1, use o seguinte comando:

```
route add default gw 10.10.15.1
```

Com isto todos os pacotes que são destinados a uma rede “externa” passarão por este endereço.

Já para remover a rota do IP 10.10.0.1, execute o seguinte comando:

```
route del 10.10.0.1
```

7.3.5.3) /bin/netstat

Mostra as conexões de rede, assim como a tabela de roteamento. Para exibir a tabela de roteamento de uma máquina também podemos utilizar o comando:

```
/bin/netstat -rn
```

Já para exibir a lista de todos os serviços TCP que estão ativos para a máquina que se está conectada

```
/bin/netstat -a | less
```

7.3.5.4) /usr/bin/ssh

Este comando visa permitir a conexão em uma máquina remota. Feita a conexão todos os comandos e ações são realizadas na máquina remota.

Para se conectar a máquina “uther” com o usuário “merlim” devemos usar o seguinte comando:

```
/usr/bin/ssh -l merlin uther
```

7.3.5.5) /usr/bin/ftp

Este comando permite a realização de downloads e uploads de arquivos pela rede.

Para realizar a conexão com o site "ftp.uther.avalon.org.br" poderíamos utilizar o seguinte comando:

```
/usr/bin/ftp ftp.uther.avalon.org.br
```

Os principais comandos que podem ser utilizados quando a conexão de FTP está aberta são:

get : baixa um arquivo do site para a máquina

mget: baixa vários arquivos do site para a máquina

put: envia um arquivo da máquina para o site

mput: envia vários arquivos da máquina para o site

hash: imprime um "#" a cada 1024 bytes

prompt: habilita / desabilita o prompt

cd: altera o diretório corrente

ls: lista o conteúdo do diretório

pwd: imprime o caminho do diretório atual

help: exibe uma ajuda sobre todos os comandos disponíveis

7.3.5.6) /bin/ping

Este comando envia uma série de pacotes para um determinado host e fica aguardando uma resposta.

Por exemplo, para enviar 10 pacotes para o host 10.10.15.1, devemos usar o seguinte comando:

```
ping -c 10 10.10.15.1
```

7.4) Rede PPP

Os modems são tidos como um dos periféricos mais simples de serem configurados, porém nem todos os modems internos funcionam perfeitamente no Linux, principalmente os Win-modems, mas o mesmo não acontece com os modems externos.

Assim como outros periféricos, o modem é ligado em uma das portas de comunicação do computador, no Linux todas a porta, assim como todo periférico físico é associado a um ou mais arquivos no diretório /dev.

No mundo DOS, por sua vez, as portas paralelas são referenciadas simplesmente como LPT1, LPT2 e assim por diante, mas na maioria dos computadores existe uma e somente uma porta paralela, já no mundo Linux, as portas paralelas correspondem arquivos de dispositivos denominados “**lp x** ”, onde x é o número da porta e neste caso a principal diferença com relação ao DOS é que a numeração das portas, no Linux, começa pelo 0 ou seja, o que é a LPT1 no DOS e na realidade é o “/dev/lp0” no Linux

As portas seriais por sua vez são associadas a dois arquivos de dispositivos ao invés de um único, o primeiro é o arquivo que controla as conexões de entrada e o outro as de saída, o arquivo que controla as conexões de saída é o /dev/ttyS x , onde x é o número da porta, mas lembre-se que a numeração das portas, no Linux, sempre começa pelo número 0, já o arquivo da porta de conexão de entrada tem caído em desuso mas ele ainda existe devido a compatibilidade com versões anteriores do Linux e pode ser localizada como /dev/cu x , onde x representa o número da porta.

Visto que já conhecemos um pouco sobre as portas de comunicação no Linux, vamos passar para o processo de instalação de um modem em um computador, para que este processo possa ter início necessitamos verificar os seguintes itens?

- ✓ O modem está conectado corretamente no Computador?
- ✓ Ele está usando uma porta serial dedicada?
- ✓ Ele está causando algum conflito de IRQ e de I/O com um outro dispositivo?

Para a instalação de um modem externo o processo é bem simples basta conectá-lo a uma das portas seriais, geralmente os usuários necessitam de no máximo duas portas deste tipo, uma para o mouse e outra para o modem, sendo que normalmente o mouse fica na COM1 e o Modem na COM.

Os modems internos necessitam ser colocados em um SLOT livre dentro do computador e ainda precisam ser configurados para usar uma porta serial específica, esta configuração pode ser feita via jumpers ou via software, sendo que o segundo caso muitas vezes não funciona no Linux, e é geralmente a terceira ou a quarta porta serial, a melhor escolha para um modem interno.

Associado a cada porta serial existe um pedido de interrupção (IRQ) e um endereço de entrada e saída (I/O), estes valores numéricos são utilizados para que seja possível a comunicação do sistema operacional com os dispositivos, a tabela a seguir possui a descrição das 4 portas seriais:

Nome DOS	Dispositivo Linux	IRQ	I/O
COM1	/dev/ttyS0	4	0x3f8
COM2	/dev/ttyS1	3	0x2f8
COM3	/dev/ttyS2	4	0x3e8
COM4	/dev/ttyS3	3	0x2e8

O fato das portas 1 e 3 assim como as portas 2 e 4 compartilharem o mesmo IRQ traz certos problemas uma vez que elas não podem ser utilizadas em conjunto, a menos que estes valores sejam alterados na BIOS, por exemplo.

Em muitos casos os modems internos permitem utilizar endereços que diferem da tabela anterior, mas isto só deve ser feito se você já tiver outros dispositivos seriais utilizando os endereços acima, para maiores informações sobre este processo verifique o Howto de dispositivos seriais.

7.4.1) Ferramentas de Configuração:

Para testarmos a configuração do Modem podemos fazer uso do aplicativo "**Minicom**", mas para tal necessitamos executá-lo a primeira vez com o parâmetro **-s**, para que seja criado o arquivo de configuração, assim sendo inicie o "Minicom" com o seguinte comando:

```
minicom -s
```

Iniciado o "minicom" realize os seguintes passos:

1. Selecione o menu **Serial port setup**;
2. Pressione "**A**" para definir o dispositivo serial, neste caso deve ser mudado a localização do modem para o arquivo de dispositivo correto;
3. Pressione "**E**" para configurar o seu modem de acordo com o sistema com o qual você ira se conectar, para velocidade deve ser selecionado o valor de 57600 bps caso você esteja utilizando um modem de 14,4 Kbps e o valor de 115200 bps para modems mais velozes;
4. Já para a opção de paridade de bits de dados deve ser selecionada a sem paridade (**letra J**), visto que a maioria das conexões atualmente são feitas sem nenhum tipo de paridade;
5. Quanto a opção Data, deve ser selecionada a opção 8 stop bits;
6. Pressione a Tecla **Enter**;
7. Selecione a opção **Save Setup as Dfl**, para salvar as configurações como padrão e em seguida selecione Exit. Isto apresentará o emulador de terminal "**minicom**".

Findo estes passos o prompt OK deve aparecer, isto significa que o modem foi configurado com sucesso. Você pode testar isso mais a fundo, para tal digite AT e pressione Enter, em seguida você deve

receber como resposta a mensagem OK

Se isto funcionar você pode tentar discar para um sistema e verificar se consegue se conectar.

7.4.2) Conectando o Linux com a Internet

Toda conexão da internet com uma máquina através de um modem é do tipo PPP, onde PPP significa Point-to-Point Protocol, que foi projetado para fornecer um método pelo qual o TCP/IP seja estendido através de uma conexão de modem analógica. É possível que uma conexão PPP use endereços fixos ou dinamicamente definidos pelo servidor, bem como conexões automáticas ou manuais e etc...

Para estabelecer uma conexão PPP com o Linux, é necessário que o kernel inclua suporte PPP, o que é um default na maioria das distribuições

No Linux são necessários dois comandos para manipular as conexões PPP, um é o “/usr/bin/pppd” e o outro é o “/usr/sbin/chat”, para verificar se eles estão instalados execute o comando:

```
rpm -q ppp
```

Caso não estejam instalados, procure no CD de instalação da sua distribuição por um arquivo com um nome parecido com `ppp-2.3.7-2.i386.rpm`, encontrado este arquivo execute o seguinte comando para instalá-lo

```
rpm -i ppp-2.3.7-2.i386.rpm
```

Uma conexão é feita e mantida pelo “pppd”, em quase todo o sistema de provedor de internet o processo de conexão ocorre da seguinte forma?

- 1) Conexão do modem;
- 2) Processo de Login;
- 3) Após a autenticação do usuário, o provedor começa a tentar estabelecer uma conexão PPP, enviando o endereço IP do cliente para o sistema do cliente.

Existem duas formas de se realizar a autenticação de usuários com o provedor de internet, uma é o de prompt-e-resposta de texto puro e a outra que usa protocolo de autenticação especial, que geralmente é o PAP.

7.4.2.1) Processo de conexão para texto puro

Para que seja possível se conectar a um ambiente configurado para autenticação de texto puro devemos fazer uso do “pppd” da seguinte forma.

Apesar do “pppd” gerenciar o dispositivo de modem, ele não é capaz de realizar a discagem e o “login”, para isto fazemos uso do programa denominado “chat”, assim, devemos utilizar um script que passe para o “chat” os parâmetros necessários para a conexão. Um exemplo de um script simples seria:

```
“ ” ATDT22339999 CONNECT “ ”
```

OBS: Logo após o ATDT deve vir o telefone do provedor, ou o telefone a ser discado.

No exemplo anterior ficou faltando introduzir o nome do usuário e a senha, para tal vamos supor que o nome do usuário seja **morgana** e a senha seja **pimenta**, com isto nosso script ficaria da seguinte forma:

```
“ ” ATDT22339999 CONNECT “ ” Username: morgana Password: pimenta
```

Depois que temos um script funcionando devemos integra-lo ao “pppd”, para isto temos que conhecer as seguintes opções do pppd:

connect: define qual será o programa utilizado para estabelecer uma conexão;

noipdefault: é utilizado quando o provedor fornece um endereço IP dinâmico, toda vez que ocorre um conexão;

defaultroute: diz ao “pppd” para incluir uma rota padrão na tabela de roteamento do sistema, usando o sistema remoto como gateway padrão, esta entrada é removida quando a conexão é desfeita.

A estrutura do comando pppd é a que se segue:

```
pppd [nome_do_device] [velocidade_do_device] [opções]
```

Logo o comando completo ficaria da seguinte forma:

```
pppd /dev/modem 57600 connect '/usr/sbin/chat “ ” ATDT22339999 CONNECT “ ” \  
Username: morgana Password: pimenta' noipdefault defaultroute
```

OBS:Tudo em uma única linha.

Conseguida a conexão devemos executar dois comandos para verificar se tudo ocorreu bem. O primeiro comando é o **“ifconfig”**, este deve retornar uma saída contendo algo semelhante a :

```
ppp0 .....  
.....  
.....
```

O seguinte comando a ser utilizado é o **“route”** para verificar se foram incluídas linhas na tabela de roteamento para criar uma rota padrão através da máquina do ISP.

7.4.2.2) Processo de conexão com uma autenticação PAP

Caso o provedor de acesso a internet possua uma autenticação do tipo PAP ,o processo de conexão muda um pouco, o nosso script fica mais simples, mas existe a necessidade da criação de um arquivo contendo todas as contas PAP, este arquivo é o **“/etc/ppp/pap-secrets”**, o formato das entradas deste arquivo é o seguinte:

nome_do_usuario	device/interface	senha
-----------------	------------------	-------

Por exemplo:

#Nome	Device	Senha
morgana	*	pimenta

O asteriscos no lugar do “device” indica que ela pode ser utilizada para realizar conexões com qualquer interface, no exemplo acima, a primeira linha é a penas um comentário, com o objetivo de ajudar ao preenchimento do arquivo de configuração.

Desta forma, a linha de comando completa, para realizar uma conexão com uma autenticação do tipo PAP, seria:

```
pppd /dev/modem 57600 connect '/usr/sbin/chat "" ATDT22339999 CONNECT "" user morgana
```

OBS: Tudo em uma única linha.

A diferença deste comando é a inserção de um parâmetro chamado “user”, que identifica qual das contas de usuário cadastradas no arquivo “pap-secrets” será utilizada na conexão.

7.4.2.3) Verificando outros arquivos necessários:

Outro fator importante no processo de conexão é verificar se os serviços de DNS estão perfeitamente configurado, caso contrário o computador não irá conseguir resolver os nomes de “sites” em endereços “IP”, para tal é necessário que os seguintes arquivos estejam configuradas da forma correta:

```
“/etc/host.conf”  
    oder host,bind  
    multi on
```

```
/etc/resolv.conf  
    search  
    nameserver 168.196.15.1
```

No caso do “resolv.conf” o endereço “168.196.15.1” deve ser substituído pelo endereço IP do servidor de nomes de seu provedor de acesso à internet, caso haja mais de um servidor novas linhas podem ser inseridas no arquivo, como no exemplo abaixo:

```
/etc/resolv.conf  
    search  
    nameserver 168.196.15.1  
    nameserver 168.196.15.2
```

7.4.3) Finalizando a conexão

Para finalizar uma conexão com a internet basta matar o processo chamado “pppd”, utilizando para isto o comando “kill”, mas devemos primeiro descobrir o PID do processo com o comando “ps”, a seguir temos o exemplo deste processo:

```
kill -9 'cat /var/run/ppp0.pid'
```

7.4.4) Script de Login e Logout

A seguir temos dois scripts que podem ser utilizados para o login e o logout de uma a conexão PPD com um provedor de acesso a internet:

login.bash

```
#!/bin/bash
/usr/sbin/pppd /dev/modem 57600 connect '/usr/sbin/chat "" ATDT<telefone> CONNECT "" '\
noipdefault defaultroute user <usuario>
```

onde:

<telefone> é o número do telefone a ser discado

<usuario> é o nome da conta de usuário cadastrada no arquivo pap-secrets

logout.bash

```
#!/bin/bash
kill -9 'cat /var/run/ppp0.pid'
```

8. Gerencia dos mecanismos de impressão

Quando solicitamos a impressão de um arquivo, nem sempre ele vai direto para a impressora, visto que a mesma pode estar sendo utilizada por um outro usuário, por uma impressão que ainda não acabou. Por este motivo, os sistemas operacionais modernos utilizam de um recurso chamado diretório de spool, onde armazenam todos os trabalhos que estão sendo encaminhados para um determinado dispositivo. De tempo em tempos a própria impressora verifica se há algum serviços neste diretório e o imprime.

Nem sempre é possível enviar um arquivo direto para a impressora, uma vez que cada impressora “fala” uma determinada linguagem de impressão, isto nos força instalar um driver de comunicação, antes de utilizar a impressora. Um driver de impressora nada mais é do que um programa que converte um arquivo qualquer, de tal forma que a impressora possa entendê-lo, ou seja, um driver de impressão funciona como um tradutor de informações.

Quando vamos instalar uma impressora devemos saber em qual porta ela será conectada. Um PC possui no máximo três portas de comunicação paralelas, no D.O.S elas são chamadas de LPT1, LPT2 e LPT3. No Linux as portas de comunicação são conhecidas como: /dev/lp0, /dev/lp1, /dev/lp2.

Contudo, a maioria dos computadores possuem apenas uma porta paralela, neste caso o dispositivo normalmente utilizado é o /dev/lp0. Para testar se a porta da impressora está funcionando perfeitamente, basta enviar um arquivo “**texto puro**” para ela, podemos fazer isto utilizando o comando abaixo:

```
cat /etc/passwd > /dev/lp0
```

8.1) Servidor de Impressão CUPS

O CUPS, que é um dos gerenciadores de impressão que podem ser utilizados no Linux. Ele pode ser utilizado via Web para: adicionar, excluir e configurar impressoras, assim como, gerenciar os serviços de impressão de uma ou de várias impressoras da rede.

Caso exista algum outro gerenciador de impressão instalado ele deve ser removido antes de que possamos utilizar o CUPS.

Considerando que o gerenciador de impressão LPR está instalado, devemos executar os seguintes comandos antes de prosseguir com a instalação do CUPS:

```
rpm -e linuxconf-printer-1.21r8.7cl  
rpm -e rhs-printfilters-1.81.4cl  
rpm -e LPRng-3.6.36.4cl
```

Feito isto, devemos localizar os seguintes pacotes:

```
cups-libs-1.1.8-3cl.i386.rpm
ghostscript-5.50-20cl.i386.rpm
ghostscript-doc-5.50-20cl.i386.rpm
ghostscript-fonts-5.50-4cl.noarch.rpm
cups-1.1.8-3cl.i386.rpm
cups-devel-1.1.8-3cl.i386.rpm
cups-devel-static-1.1.8-3cl.i386.rpm
cups-doc-1.1.8-3cl.i386.rpm
cups-drivers-1.0-2cl.i386.rpm
```

Em seguida, devemos executar os seguinte comandos para realizar a instalação dos pacotes:

```
rpm -ivh cups-libs-1.1.3-9cl.i386.rpm
rpm -ivh ghostscript-5.50-12cl.i386.rpm
rpm -ivh ghostscript-fonts-5.50-3cl.noarch.rpm
rpm -ivh ghostscript-doc-5.50-20cl.i386.rpm
rpm -ivh cups-devel-1.1.8-3cl.i386.rpm
rpm -ivh cups-devel-static-1.1.8-3cl.i386.rpm
rpm -ivh cups-doc-1.1.8-3cl.i386.rpm
rpm -ivh cups-drivers-1.0-2cl.i386.rpm
```

Concluída a instalação, precisamos iniciar dois serviços específicos um é o **HTTPD** e o segundo o próprio **CUPS**. O HTTPD é o daemon que gerencia o servidor de páginas web que está instalado na máquina e que normalmente é o “**apache**”.

```
cds
./httpd start
./cups start
```

8.2) Configurado o CUPS pelo browser

Para configurarmos o CUPS devemos iniciar um dos navegadores web que estão instalados na máquina e direcioná-lo para a porta 631 da máquina local, digitando a seguinte informação na caixa de endereço:

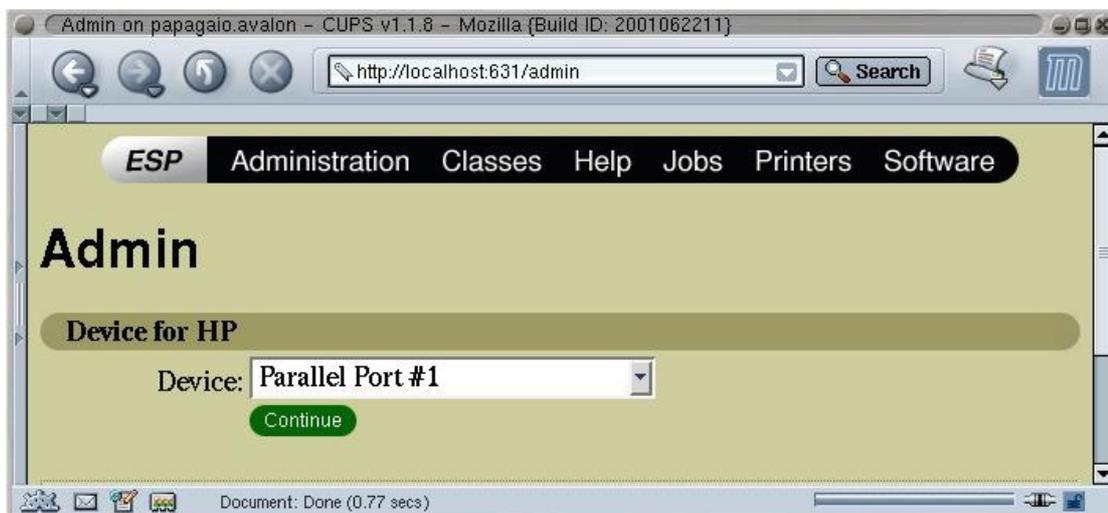
<http://localhost:631>

Para adicionar um compartilhamento de impressora em um servidor de impressão devemos seguir os seguintes passos:

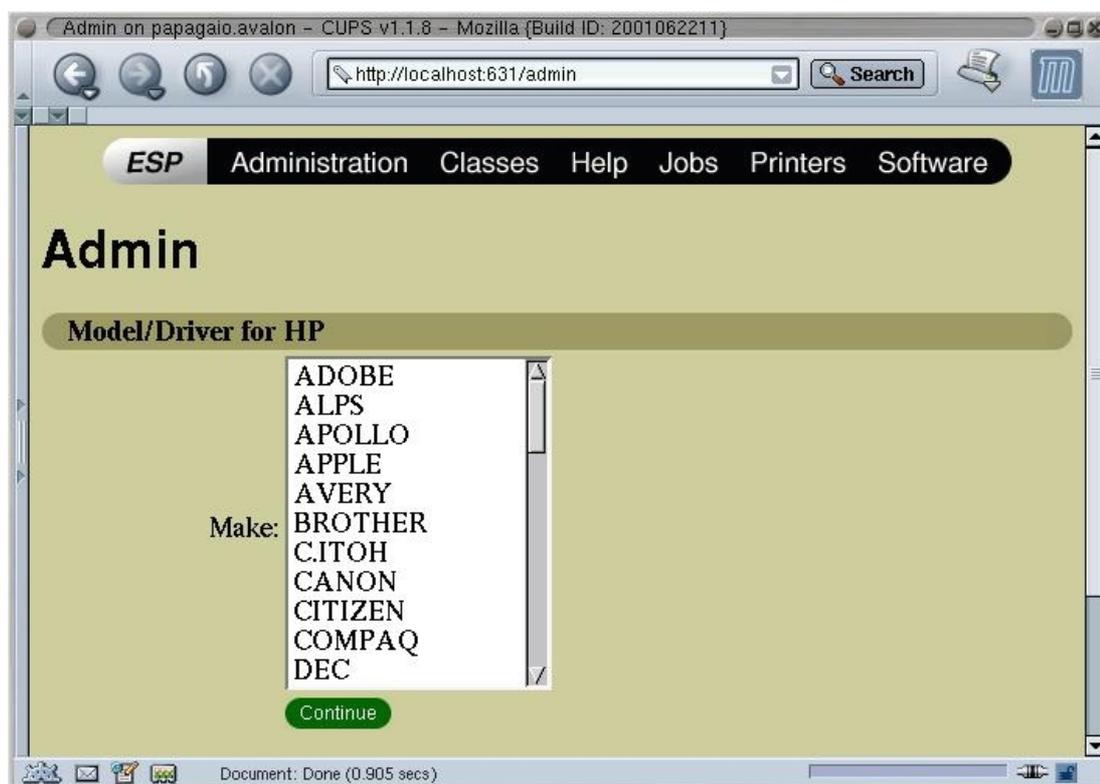
1. Clique em **Manage Printers**
2. Clique em **Add Printers**, como esta ação é restrita ao administrador, será solicitado o login name do ROOT e a senha do mesmo, informe estes valores e avance para o próximo passo.
3. Em seguida deve ser informado um nome para a impressora, a localização, que no caso é "localhost", e uma breve descrição da impressora, feito isto clique em "Continue"



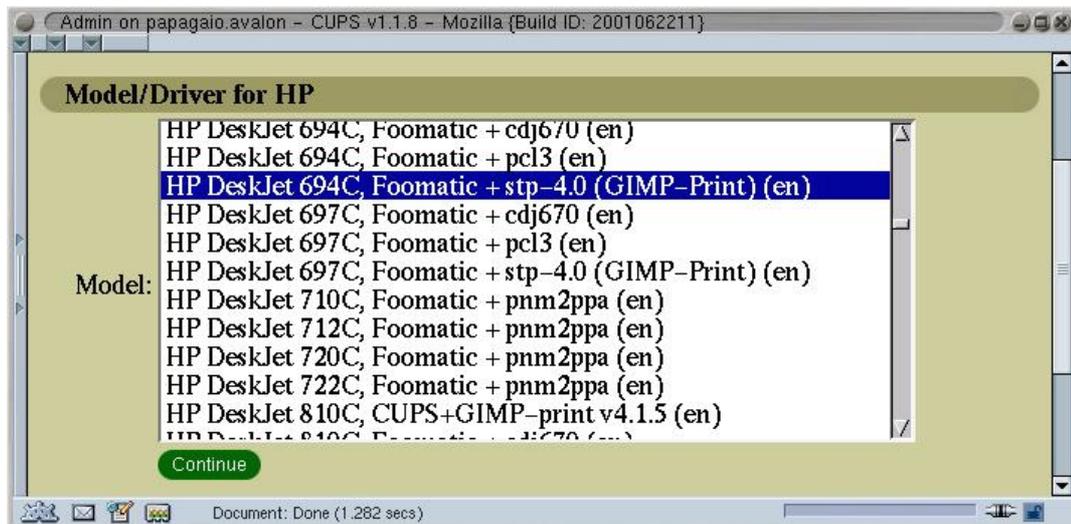
4. Em seguida, devemos selecionar a porta na qual a impressora está conectada, normalmente a opção é Parallel Port, feito isto clique em “Continue”.



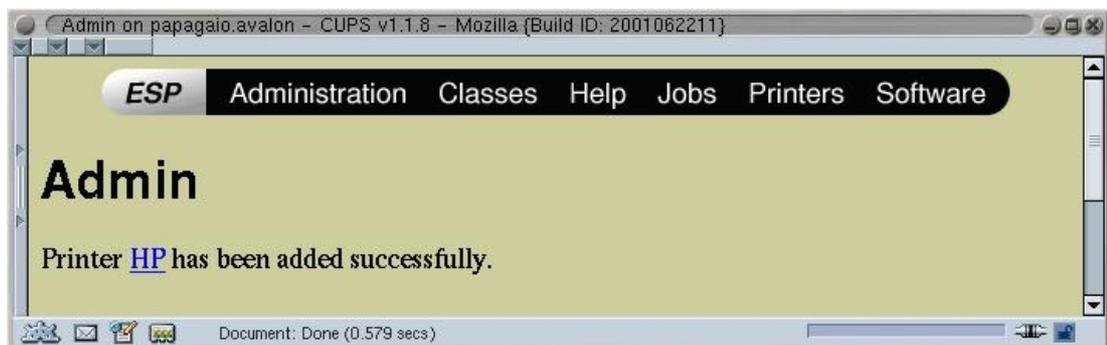
5. O próximo passo é definir qual é a marca da impressora, selecione o valor que bate com as informações sobre a sua impressora, e pressione “continue”;



6. Feito isto, será questionado quanto ao modelo da impressora, selecione-o e pressione "Continue"



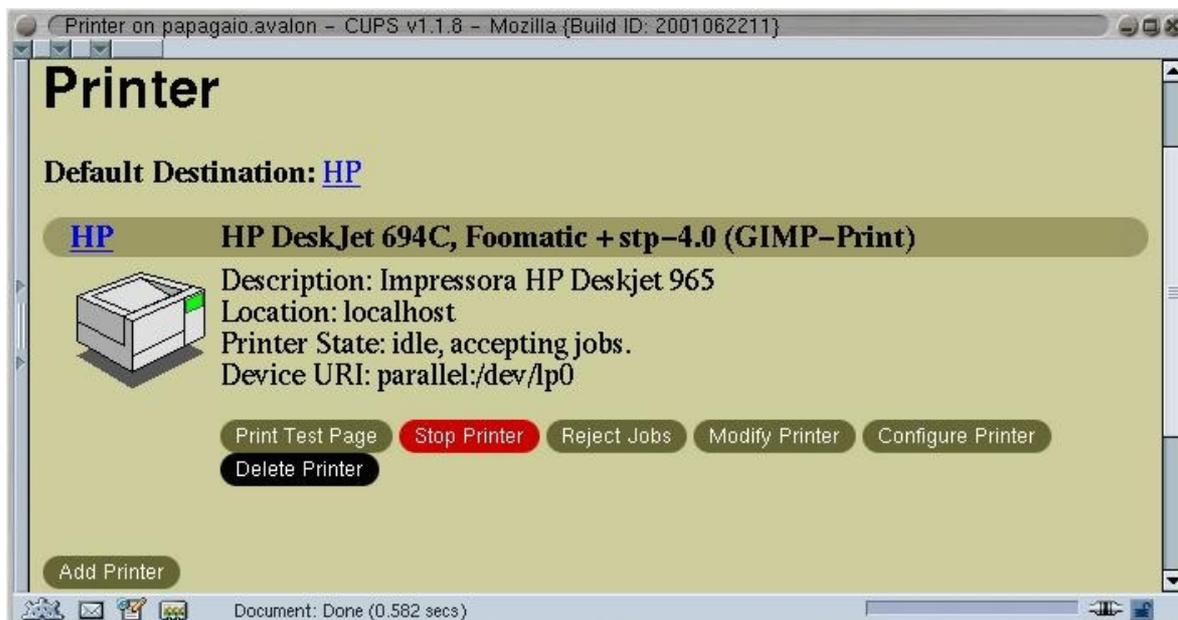
7. Em seguida surgirá uma tela informando que a impressora foi instalada com sucesso.



8. Agora para acessar esta impressora basta utilizar o comando `lpd://localhost/HP`, onde **HP** é o nome da impressora, e **localhost** é o nome da máquina.

Para a impressora que acaba de ser instalada é criado um compartilhamento, que será enviado para toda a rede, não havendo a necessidade de configurações adicionais nas demais estações da rede.

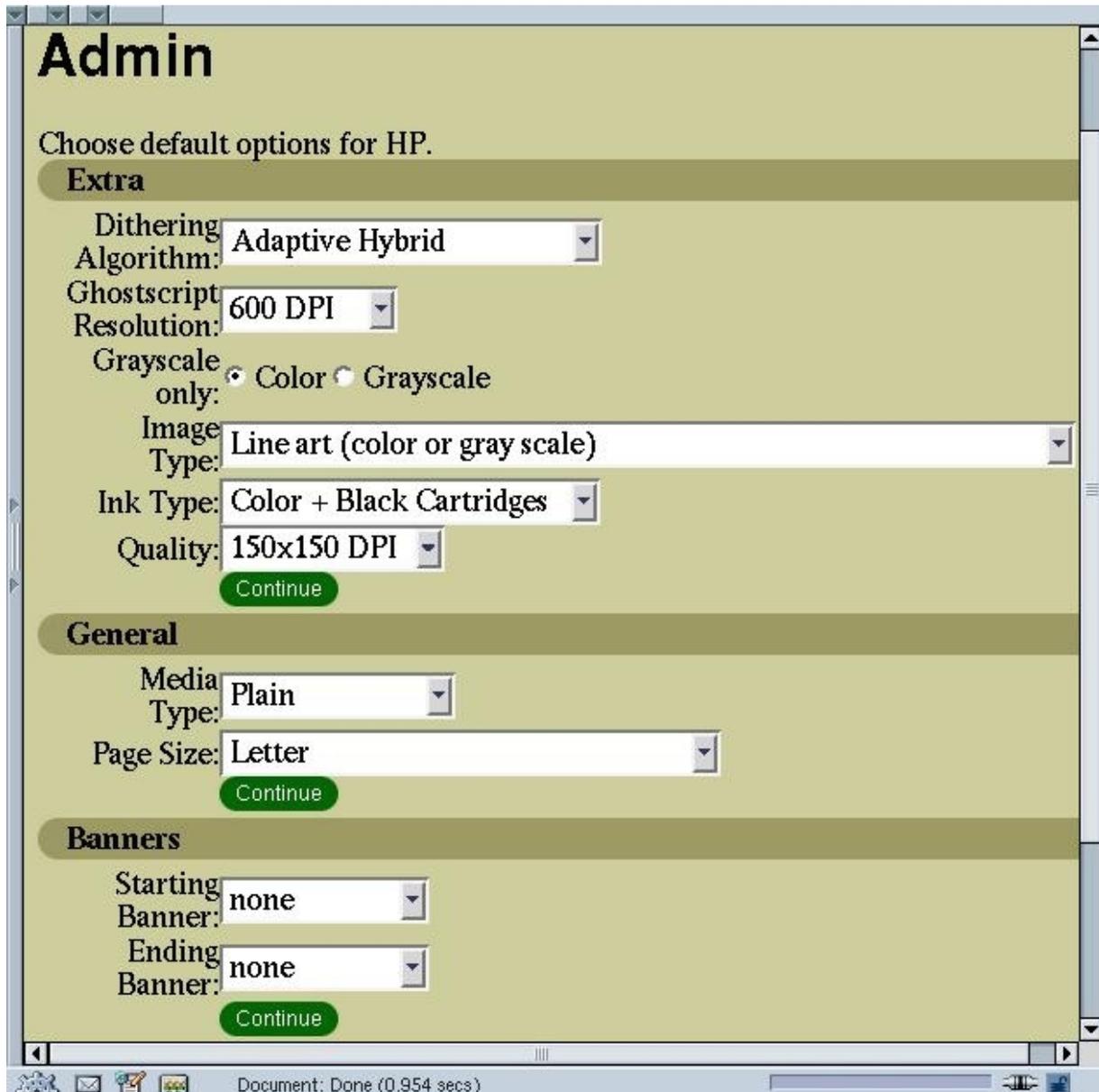
Para listar todas as impressoras compartilhadas, clique sobre a opção **Printers** localizada na barra de menu superior.



8.3) Configuração das propriedades da impressora

Existem algumas configurações que devemos realizar logo após a instalação de uma nova impressora, como é o caso do tamanho do papel, orientação, resolução e outras.

Para isto basta clicar em "Printer", na barra superior, em seguida selecione a impressora e finalmente clique "Configure Printer"



Mas lembre-se, você só pode alterar as propriedades das impressoras que estão localmente instaladas, mas todas as configurações feitas em impressoras não locais não serão aplicadas.

8.4) Configurando e Administrando o CUPS via linha de comando

Comando	Função
<i>lp</i>	Envia arquivos para impressão.
<i>cancel</i>	Cancela uma ou mais filas de impressão
<i>enable</i>	Habilita um compartilhamento de impressão.
<i>disable</i>	Desabilita um compartilhamento de impressão.
<i>lpoptions</i>	Visualiza ou configura uma impressora compartilhada (tamanho de papel, qualidade de impressão, etc)
<i>lpq</i>	Lista as filas de impressão em processamento
<i>lpr</i>	Envia arquivos para impressão.
<i>lprm</i>	Cancela uma fila de impressão
<i>lpstat</i>	Exibe informações dos compartilhamentos de impressora
<i>accept</i>	Aceita impressões vindas de um host específico
<i>reject</i>	Rejeita impressões vindas de um host específico
<i>lpadmin</i>	Administração do CUPS. Permite incluir novos compartilhamentos e outros serviços
<i>lpinfo</i>	Lista dispositivos e drivers de impressoras suportadas pelo CUPS
<i>lpmove</i>	Move uma fila de impressão para um outro destino

Exemplos de operações através da linha de comando:

Inserindo um novo compartilhamento de impressão:

```
/usr/sbin/lpadmin -p DeskJet -E -v parallel:/dev/lp0 -m deskjet.ppd
```

Removendo um compartilhamento de impressão:

```
/usr/sbin/lpadmin -x printer
```

Configurando um compartilhamento de impressão padrão:

```
/usr/sbin/lpadmin -d printer
```

Habilitando e desabilitando um compartilhamento de impressão:

```
/usr/sbin/enable printer  
/usr/sbin/disable printer
```

9.) Gerência do Ambiente Gráfico

9.1) Conceitos Básicos

O X Windows que é o ambiente gráfico do Linux e é dividido em duas camadas distintas: o Servidor X e o Gerenciador de Janela ou o Ambiente de Desktop, como podemos ver na ilustração abaixo:



Este modelo permite que os aplicativos estejam rodando em máquinas diferentes daquelas na qual ocorre a sua exibição, esta separação é algo ausente em outros sistemas como é o caso do Windows e Macintosh

9.1.1) Servidor X

O Servidor X define como será a exibição gráfica, isto é, a resolução, a intensidade da cor, as taxas de atualização e etc...

Como o Servidor X não fornece os meios necessários para o gerenciamento do comportamento dos menus, das bordas, das janelas e dos mecanismos de manipulação dos objetos gráficos, necessitamos de uma segunda camada, que é representada pelos gerenciadores de janela.

Ou seja, é de função do Servidor X realizar o suporte ao hardware e se comunicar de forma padronizada com os gerenciadores de janela, e estes, por sua vez, definem o comportamento do ambiente gráfico.

No mundo Linux existem vários servidores X, porém o mais comumente utilizado é o "**Xfree86**", isto se deve ao fato dele ser gratuito e ser encontrado na grande maioria das distribuições não comerciais do Linux.

Porém a sua configuração e instalação não são triviais, possuindo também uma performance não muito estável e imperfeita com algumas configurações de hardware. Existe um aplicativo que visa facilitar o processo de configuração do Servidor X que é o "**Xconfigurator**", que detecta automaticamente algumas configurações e em outros casos consegue apresentar algumas sugestões de configuração.

Existem outras opções de servidores X como é o caso do **Metro-X** e do **Accelerator-X**, que em muitas circunstâncias superam o **Xfree86**, porém apresentam a desvantagem de serem versões comerciais (pagas).

9.1.2) Gerenciadores de Janela

A função básica de um gerenciador de janela é realizar um refinamento na interface gráfica, que não é fornecido pelos Servidores X, entre suas funções, podemos destacar a decoração das janelas e os mecanismos para iniciar os aplicativos.

Dentre os gerenciadores de janela podemos destacar os seguintes:

- ✓ FVWM e fvwm95
<http://fvwm.math.uh.edu>
- ✓ twm
<ftp://ftp.x.org/R5contrib/vtwm-5.3.tar.gz>
- ✓ Afterstep
<http://www.afterstep.org>
- ✓ Enlightenment
<http://www.enlightenment.org>

9.1.3) Ambientes de Desktop

Estes são mais do que simples gerenciadores de janela, uma vez que, fornecem uma interface de usuário coesa e completa, que gera uma consistência na aparência dos aplicativos e das suas características.

Dentre os ambientes de Desktop podemos destacar:

- ✓ KDE – K Desktop Environment
<http://www.kde.org>
- ✓ GNOME – Gnu Neteware Object Model
<http://www.gnome.org>
- ✓ CDE – Commom Desktop Environment
<http://ww.redhat.com>
<http://www.xinside.com>

9.2) Instalação e Configuração do X Windows

Para realizarmos a instalação do X Windows faremos uso de um aplicativo que faz o gerenciamento dos pacotes do Red Hat, que é o “rpm”. Para trabalhar com ele, vamos conhecer alguns exemplos de seu uso:

1º) Instalando um novo pacote

```
rpm -i nome-do-arquivo.rpm
```

2º) Listando todos os pacotes instalados

```
rpm -qa | more
```

3º) Removendo um Pacote

```
rpm -e nome-do-arquivo
```

Conhecendo, então, o básico sobre o “rpm”, podemos iniciar o processo de instalação do nosso ambiente X. Iniciaremos pela instalação do pacote contendo os dados gerais e depois as fontes do X windows; para isto faremos uso dos seguintes comandos:

```
rpm -i XFrr86-3.3.3.1-22.i386.rpm  
rpm -i XFrr86-100dpi-fonts-3.3.3.1-49.i386.rpm  
rpm -i XFrr86-75dpi-fonts-3.3.3.1-49.i386.rpm
```

Em seguida, devemos instalar as bibliotecas necessárias para o funcionamento do servidor, para tal, utilizaremos os seguintes comandos:

```
rpm -i XFrr86-libs-3.3.3.1-49.i386.rpm  
rpm -i XFrr86-xfs-3.3.3.1-49.i386.rpm
```

O próximo passo é a instalação do utilitário de configuração, o “Xconfigurator” e o conjunto de programas do X windows, com o seguinte comando:

```
rpm -i Xconfigurator-4.1.4.i386.rpm  
rpm -i X11R6-contrib-3.3.2-6.i386.rpm
```

Finalmente, o servidor Xfree86 deve ser instalado, para este último passo devemos ter conhecimento do hardware do computador no qual estamos instalando este servidor, pois existiam vários servidores diferentes. Havia seguintes servidores disponíveis até o momento da confecção deste material:

Servidor	Hardware Suportado
Xfree86-3Dlabs	Placas de vídeo baseadas no 3D Labs
Xfree86-8514	Placas de vídeo IBM 8514 antiga ou compatível com ATI
Xfree86-AGX	Placas de vídeo baseadas em AGC, como Boca Vortex, Orchi Celsius, Spider Black Window e Hercules Graphite
Xfree86-I128	Placas de vídeo Image 128 e similar
Xfree86-Mach32	Placas de vídeo baseadas no chip Mach32 da ATI
Xfree86-Mach64	Placas de vídeo baseadas no chip Mach 64
Xfree86-Mach8	Placas de Vídeo baseadas no chip Mach 8
Xfree86-P9000	Placas de vídeo baseadas no chip Weitek, na qual a maioria das placas Diamond Viper e Orchi P9000 se enquadram
Xfree86-S3	Placas de Vídeo baseadas nos chips S3, na qual a maior das placas Diamond Stealth, Orchid Farenheits, Micro Crystal 8S.
Xfree86-S3V	Placas de vídeo baseadas no chipset S3 ViRGE
Xfree86-W32	Placas de Vídeo baseadas nos chips ET4000/W32

E os dos servidores genéricos a seguir:

Servidor	Hardware Suportado
Xfree86-VGA16	Servidor genérico de 16 cores para placas VGA
Xfree86-SVGA	Servidor genérico para dispositivos framebuffer SVGA mais simples, como as placas Cirrus Logic, Trident e etc.
Xfree86-Nomo	Servidor genérico monocromático (duas cores) para placas VGA só suportando Monitores Monocromáticos

Se você conseguir “combinar” a sua placa de vídeo e monitor com um dos servidores, instale somente o arquivo deste servidor; caso a sua configuração não se aproxime completamente de um destes servidores instale o arquivo do servidor e um dos servidores genéricos, no pior caso instale somente um dos servidores genéricos.

9.3) Configurando o X Windows

Para tal é necessário reunir certas informações sobre o seu hardware, tais como:

- Marca e modelo do monitor
- Frequência de sincronismo horizontal
- Frequência de sincronismo vertical
- Marca e modelo da placa de vídeo
- Quantidade de memória da placa de vídeo
- Tipo do mouse

Depois de obter estas informações podemos começar realmente a configurar o ambiente X.

Junto com a maioria das distribuições vêm um utilitário denominado “**Xconfigurator**”, que é bem mais simples de ser utilizado do que o “**xf86config**” que vem com o “**XFree86**”. O “**Xconfigurator**” é baseado em menus, o que facilita, e muito, o seu uso. A seguir temos um passo a passo do processo de configuração do ambiente gráfico fazendo-se uso desta ferramenta:

1. Após executarmos o comando, uma tela de boas vindas é apresentada, pressione OK para passarmos para a próxima tela;
2. Selecione a placa de vídeo que mais se assemelha a que está instalada no computador, mas uma boa escolha é a “Generic Vesa Compatible” caso você não saiba exatamente qual é a placa instalada;
3. O programa tentará detectar o monitor, caso ele não consiga selecione um, na lista de monitores disponíveis;
4. Neste passo serão detectadas a resolução e a profundidade de cores, que a placa e o monitor são capazes de suportarem. Neste ponto escolha a opção “Não detectar”;
5. Será perguntada então a quantidade de memória da placa de vídeo, informe o valor e pressione OK;
6. Lhe será questionado sobre a configuração do “Clockchip” da placa de vídeo, neste ponto selecione “Sem Configuração”;
7. Findo estes passos será realizada a ultima detecção do hardware, selecione “Detectar”;
8. Se tudo ocorrer bem, um terminal gráfico aparecerá lhe pedindo que pressione o botão OK;
9. Caso contrário aparecerá uma tela informando que um erro ocorreu. Assim, escolha “Voltar” e tente uma nova configuração.

No final deste processo é criado um novo arquivo chamado “**/etc/X11/XF86config**” contendo todas as configurações do X Windows.

Para iniciarmos o ambiente gráfico, basta utilizarmos o comando “**startx**” e um shell texto. Geralmente, somente pode haver um ambiente gráfico ativo por máquina.

10. Listas de Exercícios

Observações sobre as Lista de Exercícios:

1. As listas podem ser feitas em um grupo de no máximo 4 pessoas;
2. Elas devem ser entregues impressa ou escritas a mão;
3. Alguns das respostas já se encontram parcialmente desenvolvidas na internet, mas tomem o cuidado de alterá-las,
4. Caso utilizem um script coletado da internet, como “fonte de auxílio”, informem sobre a “fonte” utilizada,
5. A data limite para a entrega dos exercícios é a aula da semana posterior a aula na qual os mesmos foram indicados.

(1) Desenvolva um shell script para, exibir o **consumo de espaço em disco**, em múltiplos de kbytes, para cada um dos “**homedir**” existentes na máquina. Este script deve mostrar primeiro o homedir que consome menos espaço e por último os que consomem mais.

(2) Desenvolva um shell script para, realizar um backup, utilizando o comando “**tar**”. Este script deve fornecer como opções para os possíveis diretórios de origem de dados: “/”, “/bin”, “/usr”, “/boot”, “/root”, “/var”, “/etc”, “/home”, ou um que pode ser especificado pelo próprio usuário; em seguida ele deve solicitar **diretório de destino** e finalmente o **nome do arquivo** que será gerado.

(3) Desenvolva um shell script para pegar os dados que estão no **/etc/passwd**, no **/etc/shadow**, no **/etc/groups** e criar uma página web com o seguinte formato:

Listagem dos Usuários:

Nome Completo:

Username:

UID:

GID:

HomeDir:

Consumo:

Disponível:

Shell:

Senha:

Listagem dos Grupos:

Nome do Grupo:

GID:

(4) Desenvolva um Shell script para, fazer um backup dos seguintes diretórios dos Linux: “/usr”, “/var” e “/etc”. Este script deve ler o nome do arquivo a ser criado, bem como se ele deve ser compactado ou não.

OBS: utilize o comando TAR

(5) Desenvolva um Shell script para, ler dois números e uma das quatro operações matemáticas básicas e no final informar o resultado do calculo.

OBS: As operações matemáticas básicas são: +, -, * e /

(6) Desenvolva um Shell script para, ler três números e informar qual é o maior e qual é o menor.

- (7) Desenvolva um Shell script para, ler um número, calcular o somatório de 1 até o número lido e finalmente informar o resultado deste cálculo.
- (8) Desenvolva um Shell script para, ler o ano atual e a idade de uma pessoa e no final informar o ano que ela nasceu, bem como se ela é maior ou menor de idade.
- (9) Desenvolva um Shell Script, utilizando o laço de repetição For, para realizar o somatório de todos os números ímpares, de 1 até 201, e no final exibir o resultado na tela.
- (10) Desenvolva um shell script, utilizando os comandos Select e Case, para criar um menu com as seguintes opções: Somatório, Produto e Sair. Sendo que nas duas primeiras opções deve ser lido um número e em seguida realizar o cálculo selecionado no menu.
- (11) Desenvolva um shell script, utilizando o laço de repetição Until, para realizar o produto de todos os números pares de 1 até 20.
- (12) Desenvolva um shell script, para ler o nome e a idade de 20 pessoas e no final informar a idade e o nome da mais nova e da mais velha.
- (13) Desenvolva um shell script para realizar: a inclusão, a remoção de usuários, a alteração dos seus dados no /etc/passwd, da sua senha, bem como do backup do seu HomeDir
- (14) Desenvolva um shell script para realizar: a **inclusão**, a **remoção** de grupos, bem como a procura e a alteração de posse dos arquivos que pertencem a um grupo já removido.
- (15) Desenvolva um shell script para verificar quais os **IPS** estão **em uso** em uma determinada rede, para tal deve ser solicitado um IP válido e a máscara de rede, feito isto devemos verificar quais os IPs estão ativos. **Use o comando Ping**
- (16) Altere o shell anterior, para que, além de verificar os IPs válidos, seja verificado também quais os serviços estão rodando nas máquinas.
- (17) Desenvolva um shell script para, realizar um FTP (put e get) automático de arquivos, este deve solicitar o nome do **HOST**, o nome do **USUARIO**, a **SENHA**, o **TIPO** de operação (put ou get), o diretório de **ORIGEM** dos dados e o diretório de **DESTINO** dos dados.
- (18) Qual é o nome de dispositivo das portas paralelas no Linux?
- (19) Qual é o nome de dispositivo das portas seriais no Linux?
- (20) Quais são as portas seriais que não podem trabalhar em conjunto, ou seja ao mesmo tempo?
- (21) Qual é a diferença de PPP para pppd?
- (22) Qual é o comando necessário para discar para o provedor de acesso a internet, no terminal do Linux?
- (23) O que significa o "ATDT" antes do número do telefone no comando "chat"?
- (24) Desenvolva um shell script para, realizar a conexão com a internet, utilizando os aplicativos pppd e o chat. Este deve solicitar o **número do telefone** a ser discado, o **nome do usuário** e a **senha**
- (25) Desenvolva um shell script para, ler o endereço IP de uma máquina e a sua máscara de rede, em seguida informar o endereço de rede e o endereço de broadcast desta máquina.
- (26) Desenvolva um shell script para, ler um endereço IP, uma máscara de rede e um endereço de broadcast, em seguida atribuir estes valores a interface de rede chamada eth0:10.

- (27)Desenvolva um shell script para, exibir a tabela de roteamento da máquina.
- (28)Desenvolva um shell script para, ler um endereço IP e adiciona-lo como sendo a rota de saída default de gateway.
- (29)Desenvolva um shell script para, listar todos os serviços ativos na máquina.
- (30)Desenvolva um shell script para, ler o nome de uma máquina e o nome de um usuário, em seguida realizar uma conexão ssh nesta máquina com o nome do usuário informado.
- (31)Desenvolva um shell script para, ler um endereço IP e uma quantidade de pacotes, depois utilizar o comando PING para enviar esta quantidade de pacotes para o endereço lido.
- (32)Desenvolva um shell script que utilize os comandos “pppd” e “chat” de tal forma que:
- O número IP seja atribuído automaticamente;
 - O endereço IP do roteador do provedor seja adicionado a tabela de roteamento da máquina, de forma automática
- Além disto o shell script deve pedir que o usuário informe:
- A velocidade da conexão
 - O telefone do provedor de acesso a internet
 - O login name do usuário
 - A senha do usuário
- (33)Quantas e quais são as camadas nas quais podemos dividir o ambiente gráfico do Linux?
- (34)Quais são os três comandos que podem ser utilizados para configurar o ambiente gráfico?
- (35)Quais são os programas alternativos para o Xfree86?
- (36)Qual a diferença de um Gerenciador de Janelas para um Ambiente de Desktop?
- (37)Quando fazemos a instalação do Xfree86 instalamos 3 conjuntos de pacotes quais são eles?
- (38)Para que serve o comando rpm?
- (39)Qual é o arquivo de configuração do Xfree86?
- (40)Qual é o comando utilizado para iniciar o ambiente, independentemente da distribuição utilizada?
- (41)Desenvolva um shell sript para, instalar automaticamente todos os arquivos necessários para o funcionamento do Xfree86.
- (42)Desenvolva um shell script para, realizar a adição, remoção das contas de usuários, bem como a remoção de seus dados. Para tal use os comandos Select e Case para a construção e gerencia das opções do menu.
- (43)Desenvolva um shell script para, fazer o backup e o restore dos dados de uma determinada conta de usuário. Para tal use os comandos Select e Case para a construção e gerencia das opções do menu.

(44)Desenvolva um shell script para, adicionar e remover grupos de usuários. Para tal use os comandos Select e Case para a construção e gerencia das opções do menu.

(45)Desenvolva um shell script para, verificar quanto de espaço em disco um determinado diretório está ocupando e quando há de espaço livre nos sistemas de arquivo. Para tal use os comandos Select e Case para a construção e gerencia das opções do menu.

(46)Desenvolva um shell script para, exibir na tela qual o nível de execução padrão, no qual o linux inicia sua operação. Para tal use os comandos grep e cut

(47)Desenvolva um shell script, para realizar a checagem de um determinado sistema de arquivo.

10.Referências

10.1) Livros Indicados

1. Danesh, Arman; *Dominando o Linux – Red Hat Linux 6.0, A Bíblia*; Ed. Makron Books
2. Welsh, Matt; *Dominando o Linux*; Ed. Ciência Moderna
3. Siever, Ellen; *Linux, O Guia Essencial*; Editora Campus

10.2) Sites Indicados – Sobre Linux

1. Conectiva Linux: <http://www.conectiva.com.br>
2. The Slackware Linux Project : <http://www.slackware.org> <http://www.slackwarebrasil.org>
3. Linux Security Brazil: <http://www.linuxsecurity.com.br>

10.3) Sites Indicados – Busca

1. Google – Linux : <http://www.google.com.br/linux>
2. Google – UNIX-BSD : <http://www.google.com.br/bsd>