

CELSO MASSARU KAWASHIMA

Multi-Terminais Linux

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes *Linux*, para obtenção do título de Especialista em Redes *Linux*.

Orientador

Prof. MSc. Herlon Ayres Camargo

LAVRAS

Minas Gerais – Brasil

2006

CELSO MASSARU KAWASHIMA

Multi-Terminais Linux

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes *Linux*, para obtenção do título de Especialista em Redes *Linux*.

Aprovada em 29 de Setembro de 2006.

---

Prof. MSc. Simone Markenson Pech

---

Prof. MSc. Joaquim Quinteiro Uchôa

---

Prof. MSc. Herlon Ayres Camargo

LAVRAS

Minas Gerais – Brasil

2006

## DEDICATÓRIA

*A meus pais que sempre estiveram tão presentes em minha vida e me proporcionaram a base para prosseguir, à minha esposa pelo amor, companheirismo e apoio irrestrito e às minhas filhas pela compreensão e carinho.*

## AGRADECIMENTOS

*A todos que de alguma forma ajudaram-me a transpor mais essa etapa, em especial ao meu amigo de jornada Fernando José da Silva e ao meu orientador Prof. Herlon Ayres Camargo.*

## RESUMO

Apresentar a utilização da técnica de Multi-Terminais *GNU/Linux* na montagem de laboratórios de informática de escolas públicas, telecentros, centros comunitários e entidades que trabalham para diminuição da exclusão digital.

## SUMÁRIO

LISTA DE FIGURAS.....	8
LISTA DE TABELAS.....	9
APÊNDICES.....	10
1. INTRODUÇÃO.....	11
2. SOCIEDADE DA INFORMAÇÃO.....	13
2.1. Introdução.....	13
2.2. Exclusão Digital .....	14
2.3. Inclusão Digital .....	15
2.4. Telecentros.....	16
2.5. Multi-Terminais <i>GNU/Linux</i> .....	17
3. IMPLEMENTAÇÃO.....	21
3.1. Servidor utilizado.....	21
3.2. Periféricos utilizados.....	22
3.3. <i>Software</i> utilizados.....	22
3.4. Montagem inicial do <i>hardware</i> .....	23
3.5. Instalação do <i>hardware</i> referente aos demais terminais.....	23
3.6. Alteração no <i>kernel</i> .....	25
3.7. Pacotes necessários para alterar o <i>kernel</i> .....	26
3.8. Aplicação do <i>patch</i> e compilação do <i>kernel</i> .....	26
3.9. Alteração de fonte de letra padrão do <i>Fedora Core 1</i> .....	29
3.10. Criação de dispositivos necessários para o Multi-Terminais.....	29
3.11. Instalação do pacote <i>devfsd</i> .....	30
3.12. Instalação de <i>driver</i> para placas de vídeo.....	31
3.13. <i>Boot</i> com o novo <i>kernel</i> .....	31
3.14. Identificação do endereçamento de teclados e <i>mouses</i> .....	32

3.15. Utilizando <i>hotplug</i> com <i>input.agent</i> e <i>input.rc</i> .....	33
3.16. O serviço <i>ruby_init</i> .....	34
3.17. Configuração do serviço de vídeo.....	36
3.18. Testes iniciais.....	42
4. ANÁLISE DE CUSTOS .....	46
5. CONCLUSÃO.....	48
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	50
APÊNDICES.....	53

## LISTA DE FIGURAS

Figura 2.1 – Telecentro Nova Cachoeirinha.....	17
Figura 3.1. Placas de vídeo instaladas .....	24
Figura 3.2 – Ligação de teclado e <i>mouse</i> .....	24
Figura 3.3- Conexão dos cabos de vídeo, teclados e <i>mouses</i> .....	25
Figura 3.4 – Conteúdo de <code>/boot/grub/grub.conf</code> .....	28
Figura 3.5 – Conteúdo original de <code>/etc/sysconfig/i18n</code> .....	29
Figura 3.6 – Conteúdo alterado de <code>/etc/sysconfig/i18n</code> .....	29
Figura 3.8 – Conteúdo do arquivo <code>/etc/hotplug/mouse.conf</code> .....	34
Figura 3.9 – Conteúdo do arquivo <code>/etc/hotplug/kbd.conf</code> .....	34
Figura 3.10 – Conteúdo do arquivo <code>/etc/sysconfig/ruby.conf</code> .....	36
Figura 3.11 – Endereços das placas de vídeo.....	38
Figura 3.12 – Exemplo de arquivo de configuração de vídeo <code>XFreeConfig1</code> ..	39
Figura 3.13a – Primeira parte do conteúdo do arquivo <code>/etc/X11/gdm/gdm.conf</code> .	40
Figura 3.13b – Segunda parte do conteúdo do arquivo <code>/etc/X11/gdm/gdm.conf</code> ..	41
Figura 3.13c – Terceira parte do conteúdo do arquivo <code>/etc/X11/gdm/gdm.conf</code> ..	42
Figura 3.14 – Terminais apresentando tela de <i>login</i> .....	43
Figura 3.15 – Terminais em uso.....	43
Figura 3.16 – Processos referentes aos terminais e diferentes usuários.....	44



## **LISTA DE TABELAS**

Tabela 4.1 – Preços de itens de <i>hardware</i> adicionais .....	46
Tabela 4.2 – Gasto na aquisição de microcomputadores.....	47
Tabela 4.3 – Gastos na montagem de Multi-Terminais.....	47

## APÊNDICES

Apêndice A - Listagem do arquivo <code>/boot/grub/grub.conf</code> .....	53
Apêndice B - Listagem do arquivo <code>/proc/bus/input/devices</code> .....	54
Apêndice C - Listagem do arquivo <code>/etc/hotplug/input.agent</code> .....	56
Apêndice D - Listagem do arquivo <code>/etc/hotplug/input.rc</code> .....	60
Apêndice E - Listagem do arquivo <code>/etc/rc.d/init.d/ruby_init</code> .....	64
Apêndice F - Listagem do arquivo <code>/etc/sysconfig/ruby.conf</code> .....	71
Apêndice G - Listagem do arquivo <code>/etc/X11/XFreeConfig1</code> .....	73
Apêndice H - Listagem do arquivo <code>/etc/X11/gdm/gdm.conf</code> .....	75
Apêndice I - Listagem completa de processos em execução.....	88

# 1. INTRODUÇÃO

A montagem de um laboratório de informática envolve gastos na aquisição de computadores, equipamentos de rede e de *software*. Os elevados custos na aquisição de *hardware* e *software* podem ser em determinadas situações um dos principais obstáculos na montagem de laboratórios em escolas, telecentros e projetos que visam combater a exclusão digital.

A utilização de *software* livre, como o sistema operacional *GNU/Linux* e pacote de escritório como OpenOffice.org<sup>1</sup>, amenizam os gastos no quesito *software*, mas os gastos com os computadores e equipamentos de rede continuam.

Existem soluções que possibilitam a economia financeira na implementação de *hardware* e que podem ser escolhidas conforme os recursos financeiros e de *hardware* já existentes. Uma solução interessante quando já existem recursos de equipamentos mesmo que obsoletos é a implementação de LTSP<sup>2</sup>. No caso de investimento em uma estrutura nova de *hardware*, o uso de Multi-Terminais *GNU/Linux* torna-se uma opção extremamente interessante devido a economia obtida em sua implementação e com gastos de manutenção.

A busca por uma solução tecnológica que permita um uso mais racional do hardware e uma economia na implementação de laboratórios de informática, foi a motivação para o desenvolvimento deste trabalho.

---

1 OpenOffice.org - <http://www.openoffice.org.br/>

2 LTSP – GNU/Linux Terminal Server Project. Disponível em [www.ltsp.org](http://www.ltsp.org)

No capítulo 2, são apresentados fatores que levam a exclusão digital e exemplo de estrutura que possibilita a oportunidade de combate a esse problema. Mostra o conceito de funcionamento do Multi-Terminal *GNU/Linux*.

O capítulo 3, trata da montagem do *hardware* e da implementação de *software* utilizados.

No capítulo 4, é apresentado um comparativo de custo na implementação de um sistema Multi-Terminal com uma solução padrão baseada em *desktop*.

O capítulo 5, apresenta as conclusões obtidas com o trabalho.

Não é objetivo deste trabalho medir o desempenho desta solução, ou mesmo comparar seu desempenho com o de outras técnicas existentes. O objetivo é apresentar Multi-Terminais *GNU/Linux* como uma alternativa de baixo custo na montagem de laboratórios.

## 2. SOCIEDADE DA INFORMAÇÃO

### 2.1. Introdução

Vivemos hoje na Sociedade da Informação onde as TIC's<sup>1</sup> estão proporcionando às pessoas e às organizações meios diversos de comunicação e acesso às informações. Através da *Internet*, que é um dos principais representantes das TIC's, tem-se acesso a um universo de informações sobre os mais diversos assuntos. Através de mecanismos de busca como o *Google*<sup>2</sup>, informações armazenadas fisicamente em computadores que podem estar localizados em qualquer lugar do planeta e que estão conectados a *Internet*, podem ser acessadas instantaneamente.

Segundo SEABRA (2006), a *Internet* na educação traz um potencial inovador ímpar, pois permite superar as paredes da sala de aula, com a troca de idéias entre alunos de outras cidades e países, intercâmbio entre os educadores, nacional e internacionalmente, pesquisa *online* em bancos de dados, assinatura de revistas eletrônicas e o compartilhamento de experiências em comum. Este novo ambiente de aprendizagem, não reside mais apenas na escola, mas também nos lares e nas empresas.

---

<sup>1</sup> TIC's – Tecnologias de Informação e Comunicação

<sup>2</sup> Google – Mecanismo de busca na *Internet*. Disponível em [www.google.com.br](http://www.google.com.br)

## 2.2. Exclusão Digital

Atualmente, o acesso as TIC's e a *Internet* em particular é privilégio de poucos. Esse restrito acesso a *Internet* segundo MACADAR (2006), revela uma grande distância entre nações ricas e pobres. Praticamente metade das pessoas que tem acesso a *Internet* no mundo estão na América do Norte e Europa, no Brasil somente 14,1% da população utiliza a *Internet*, valor baixo comparado com nossos vizinhos como Argentina (26,4%), Chile (35,7%) ou mesmo Uruguai (20,8%) IWS (2006). A sociedade rica usa com intensidade as redes informacionais para se comunicar, armazenar e processar informações, enquanto os países pobres e em desenvolvimento têm suas populações distantes dos benefícios destas redes.

O termo normalmente utilizado para caracterizar a distância entre os indivíduos, organizações e comunidades que têm acesso às chamadas TIC's (*Internet* principalmente), e aqueles que não o têm é "*Exclusão Digital*". MACADAR (2006) e FILHO (2003), afirmam que um dos fatores que leva à exclusão digital é a exclusão econômica, pois a falta de recursos financeiros inviabilizaria o acesso as TIC's existentes. Sob esse prisma, as camadas financeiramente carentes da população estariam condenadas a serem rotuladas de excluídos digitais. FILHO (2003), enfatiza que a exclusão digital aprofunda a exclusão sócio-econômica.

Outros fatores além do financeiro colaboram para a existência dos excluídos digitais, tais como o baixo nível de educação das pessoas, a resistência natural de determinadas pessoas ao uso das TIC's, o idioma utilizado na

confeção das páginas existentes na *Internet* que são na maioria em Inglês MACADAR (2006).

Segundo SEABRA (2006), uma sociedade baseada cada vez mais na troca de valores simbólicos, do dinheiro à informação, vai mudar o eixo da economia, acabar com o conceito atual de trabalho, valorizar mais que tudo o conhecimento e a aprendizagem. Neste cenário, os excluídos serão cada vez mais excluídos com o poder se concentrando nas esferas virtuais (com profundo controle nas esferas reais), a não ser que se implementem eficazes e massivas ações para promover sua "*inclusão digital*".

### **2.3. Inclusão Digital**

Inclusão digital é a denominação dada aos esforços que visam proporcionar aos excluídos digitais a obtenção dos conhecimentos necessários para utilizar com um mínimo de proficiência os recursos das TIC's e ter acesso regular a estes SABACK (2005). REBÊLO (2005) vai além quando diz que incluir digitalmente não é apenas alfabetizar a pessoa em informática, mas também melhorar os quadros sociais a partir do manuseio dos computadores. Não apenas ensinar o “bê-á-bá” da informática, mas mostrar às pessoas como ganhar dinheiro e melhorar de vida com ajuda das TIC's. Exemplifica esse tipo de ação com o caso de uma ONG em Honduras, que instalou estações de trabalho em comunidades rurais e treinou as pessoas a usarem as ferramentas e como tirar proveito delas – os agricultores e artesões começaram a vender seus trabalhos pela *Internet* e jovens da comunidade usam salas de bate-papo para ensinar espanhol a europeus .

Para que o processo de inclusão digital tenha sucesso, três pontos fundamentais devem ser trabalhados, a existência das TIC's, renda e educação FILHO (2003).

## 2.4. Telecentros

Como o acesso as TIC's é essencial em um programa de inclusão digital, governos e organizações tem investido na criação de instalações comunitárias onde os excluídos digitais (principalmente os carentes financeiramente), podem ter acesso gratuito aos recursos tecnológicos e podem aprender a utilizá-los. Esses locais são chamados genericamente de Telecentros.

Um Telecentro é um lugar físico, de fácil acesso público, que oferece gratuitamente serviços de informática e telecomunicações, num contexto de desenvolvimento social, econômico, educacional e pessoal. Sua concepção se baseia na crença de que "*o cidadão tem o seu poder aumentado quando tem acesso ao conhecimento*", LITTO (2000).

Telecentros têm muitas conseqüências positivas e estrategicamente importantes para o cidadão: (1) acesso fácil à informação necessária para o cidadão levar a vida para frente com dignidade; (2) oportunidades para fortalecer a capacitação profissional dos cidadãos através de educação local e a distância; (3) aumento das oportunidades para auto-expressão local. Na figura 2.1 é apresentado como exemplo a estrutura física do Telecentro Nova Cachoeirinha<sup>1</sup> em São Paulo.

---

<sup>1</sup> Fonte: [http://www.telecentros.sp.gov.br/noticias/zona\\_norte/vila\\_nova\\_cachoeirinha/index.php?p=3406](http://www.telecentros.sp.gov.br/noticias/zona_norte/vila_nova_cachoeirinha/index.php?p=3406)





Figura 2.1 – Telecentro Nova Cachoeirinha

Os custos na montagem dos telecentros podem ser minimizados com a utilização de *software* livre, uma vez que não existem gastos na sua aquisição e podem ser utilizadas estruturas mais econômicas de *hardware* conforme mostra PORTAL (2006).

## 2.5. Multi-Terminais *GNU/Linux*

Terminal pode ser definido como um conjunto de dispositivos (monitor de vídeo, teclado e *mouse*) que permitem a interação entre o usuário e o computador. Em uma estrutura de multiterminal, vários conjuntos de monitores de vídeo, teclados e *mouses* são conectados a uma mesma *CPU* e uma camada de *software* controla a independência de funcionamento desses conjuntos, MULTITERMINAL (2006).

Multi-Terminais *GNU/Linux* é o resultado de trabalhos desenvolvidos pela comunidade que trabalham em prol do *software* livre e que possibilitam a utilização de uma mesma *CPU*<sup>1</sup> por diversos usuários que irão compartilhar a sua capacidade de processamento. Esse tipo de implementação faz uso da capacidade de processamento dos microcomputadores atuais, que atendendo somente um usuário por vez, fica boa parte do tempo ocioso.

No quesito *hardware*, os teclados e *mouses* são conectados a *CPU* através de portas PS/2 e USB e para cada monitor de vídeo deve ser instalada uma placa de vídeo adicional. A quantidade de terminais disponíveis é limitada principalmente pela quantidade de *slots* tipo PCI existentes, nos quais são conectadas as placas de vídeo adicionais.

No *GNU/Linux*, a interface gráfica que normalmente é utilizada para interação com o usuário baseia-se em uma estrutura cliente-servidor que é o *X Window System*. Os servidores do sistema *X Window System* são executados em computadores com *Displays* baseados em mapas de *bits*. O servidor distribui as ações de entrada do usuário (*mouse* e teclado) e aceita os pedidos de saída através de vários programas clientes, X (2006). Nessa estrutura um *Display* é atribuído a um único usuário, no caso de multi-terminal é necessário um *Display* para cada um dos usuários conectados.

O *XFree86*<sup>2</sup>, que é uma implementação do *X Window System* não oferece suporte a múltiplos *Displays*, utiliza o modelo de entrada de dados baseado em terminais virtuais que é implementado através de *software*

---

1 *CPU* - Central Processing Unit, ou Unidade Central de Processamento

2 *XFree86* – [www.xfree86.org](http://www.xfree86.org)

simulando um *tty*. O *kernel Linux* suporta vários terminais virtuais, mas somente um deles recebe eventos de teclado por vez. Caso mais de um teclado esteja conectado ao computador, os eventos de todos os teclados são enviados ao terminal virtual ativo. Isso impede a execução concorrente de dois ou mais servidores *X*, pois apenas um pode estar ativo num determinado momento, mesmo que utilizem *hardware* de vídeo distintos, MULTITERMINAL (2006).

Uma proposta para permitir a utilização de uma mesma *CPU* por mais de um usuário foi apresentada por Miguel Freitas, FREITAS (2006). Essa implementação utilizava dois binários do servidor *XFree86* sendo executados simultaneamente no computador, um dos binários foi alterado de forma a acessar um teclado que não fosse o terminal virtual do *kernel*. Nesse caso, somente 2 terminais são disponibilizados para uso. FREITAS (2006), recomenda para essa implementação: “*A placa de vídeo não pode ser primária, o mouse não pode ser o mesmo usado pelo gpm e o teclado deve ser USB e não deve gerar scancodes no console normal do Linux*”. O problema desta solução é a incompatibilidade das placas de vídeo, devido a concorrência no acesso ao barramento PCI, MULTITERMINAL (2006).

Baseado na solução desenvolvida por FREITAS (2006), foi desenvolvido um *patch* para o *kernel*, denominado *Backstreet Ruby* RUBY (2006) que cria terminais virtuais adicionais, cada um dos terminais virtuais criados é associado a um teclado existente no sistema. Isso permite que sejam executadas diversas instâncias do *XFree86*, uma para cada terminal virtual. A quantidade de terminais virtuais a serem criados é definido através do parâmetro *dumbcon=<qtde de terminais>* que é passado ao *kernel*.

Esta solução que é adotada como solução para implementação de Multi-Terminais neste TCC, apresenta uma boa estabilidade com placas de vídeo *NVIDIA*<sup>1</sup> e segundo MULTITERMINAL (2006), proporciona boa estabilidade com placas de vídeo *SIS315*<sup>2</sup> e outras *chipsets*<sup>2</sup> desde que sejam primárias no sistema.

As desvantagens dessa solução são o problema de concorrência no barramento PCI entre as placas de vídeo e a necessidade de um *patch* específico para cada versão de *kernel*.

---

1 Nvidia - [www.nvidia.com](http://www.nvidia.com)

2 SIS – [www.sis.com](http://www.sis.com)

## 3. IMPLEMENTAÇÃO

A implementação de Multi-Terminais *GNU/Linux* que é o objeto desta monografia, utiliza a distribuição *Fedora Core 1* e contempla quatro terminais. Está em uso a 2 anos como plataforma de desenvolvimento de projetos e treinamento em linguagens PHP e Java, utilizando *software* como *BlueFish*<sup>1</sup>, *Eclipse*<sup>2</sup>, *Firefox*<sup>3</sup>, *Apache*<sup>4</sup> e banco de dados *MySQL*<sup>5</sup>.

### 3.1. Servidor utilizado

O computador utilizado no projeto possui a seguinte configuração:

- Placa mãe marca ASUS<sup>6</sup> modelo A7V8X
- Processador Athlon<sup>7</sup> XP 2.0 Ghz
- Memória RAM de 768 MB
- *Hard Disk* marca Seagate<sup>8</sup> de 80 GB 7.200 RPM
- Leitor de CD-ROM marca LG<sup>9</sup>

---

1 BlueFish – <http://bluefish.openoffice.nl/>

2 Eclipse – [www.eclipse.org](http://www.eclipse.org)

3 Firefox - <http://br.mozdev.org/firefox/>

4 Apache – [www.apache.org](http://www.apache.org)

5 MySQL – [mysql.com](http://mysql.com)

6 ASUS – [www.asus.com](http://www.asus.com)

7 Athlon – [www.amd.com](http://www.amd.com)

8 Seagate – [www.seagate.com](http://www.seagate.com)

9 LG – [www.lge.com](http://www.lge.com)

### 3.2. Periféricos utilizados

Foram utilizados os seguintes periféricos para configurar os 4 terminais:

- 3 Placas de vídeo tipo PCI marca NVIDIA modelo RIVA TNT2
- 1 Placa de vídeo tipo AGP marca NVIDIA modelo RIVA TNT2
- 4 Mouses tipo 2 botões com *plug* tipo PS/2
- 4 Teclados tipo ABNT2 com *plug* tipo PS/2
- 3 Conversores tipo entrada USB para 2 saídas PS/2
- 4 Monitores de vídeo marca Samsung<sup>1</sup> modelo 591V

### 3.3. Software utilizados

Os *software* utilizados na configuração do Multi-Terminais *GNU/Linux*:

- Distribuição *Fedora Core 1*<sup>2</sup>
- Fontes do *kernel*, obtido do arquivo *linux-2.4.26.tar.bz2*
- *Patch* Backstreet Ruby, obtido do arquivo *bruby-2.4.26-JD-20040626.diff.bz2*
- Software *devfsd*, obtido do arquivo *devfsd-2.4.3-12.i386.rpm*
- *Driver* para placa de vídeo, obtido do arquivo *NVIDIA-Linux-x86-1.0-6111-pkg1.run*
- *Xfree86* modificado, obtido do arquivo *Xfree86-430-prefbusid3.bz2*

Os *sites* onde os *software* utilizados foram obtidos, serão citados no

---

<sup>1</sup> Samsung – [www.samsung.com](http://www.samsung.com)

<sup>2</sup> Fedora - <http://fedora.redhat.com/>

texto que trata da instalação e configuração destes.

### **3.4. Montagem inicial do *hardware***

Inicialmente, a configuração do hardware é a de um computador comum, o conjunto de *mouse* e teclado conectados nas entradas PS/2 existentes na *CPU* e a placa de vídeo tipo AGP instalada no *slot* correspondente existente na placa mãe. Deve-se configurar no *setup* da máquina que a *interface* de vídeo principal é a AGP, isso implica que essa será a primeira a ser ativada quando o Multi-Terminais for ligado e será também o terminal principal do sistema.

### **3.5. Instalação do *hardware* referente aos demais terminais**

Os conjuntos de placas de vídeo, teclados e *mouses* referentes aos demais terminais são instalados, mas só poderão ser utilizados após as configurações serem efetuadas. As placas de vídeo devem ser instaladas nos *slots* PCI, conforme ilustrado na figura 3.1.



Figura 3.1. Placas de vídeo instaladas

Teclados e *mouses* ligados através do adaptador USB/PS-2, conforme pode ser visto na figura 3.2. A utilização do adaptador USB/PS-2 foi escolhida devido ao menor custo de aquisição de teclados e *mouses* com conector PS-2, em relação aos mesmos dispositivos com conexão USB.

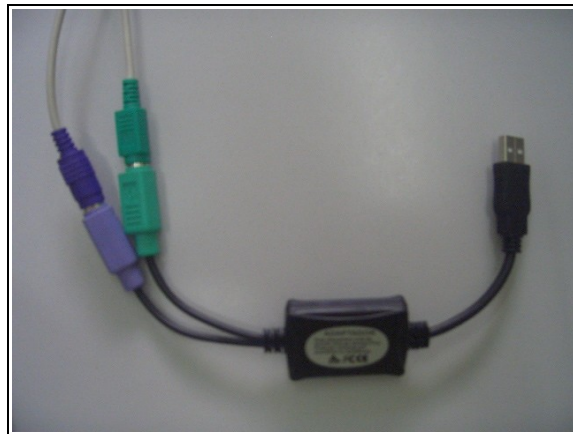


Figura 3.2 – Ligação de teclado e *mouse*



A conexão dos conjuntos de teclados e *mouses* a *CPU*, é apresentada na figura 3.3. Nesta figura, também pode ser observado a ligação dos cabos dos monitores de vídeo nas respectivas placas de vídeo.

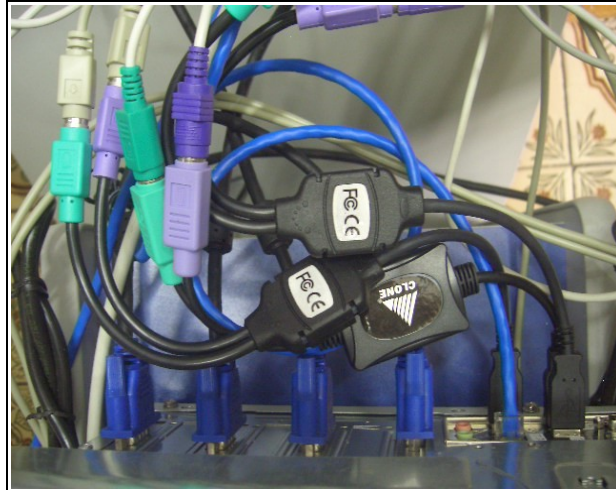


Figura 3.3- Conexão dos cabos de vídeo, teclados e *mouses*

### 3.6. Alteração no *kernel*

O *kernel* a ser utilizado no Multi-Terminais, deve permitir o funcionamento de conjuntos independentes de teclados e *mouses* que serão utilizados em cada um dos terminais. Para que isso seja possível, é necessário efetuar uma alteração no *kernel*, através da aplicação de um *patch* que tem como função reorganizar a entrada, o console e o subsistema de *framebuffer* do *kernel*, permitindo assim o controle de diversos consoles.

### 3.7. Pacotes necessários para alterar o *kernel*

O *patch* a ser aplicado é o Backstreet Ruby, que pode ser obtido em <http://disjunkt.com/dualhead/#links> . Nessa implementação foi utilizado o arquivo *bruby-2.4.26-JD-20040626.diff.bz2*, que contém o *patch* a ser utilizado.

Como um *patch* é desenvolvido para uma versão específica de *kernel*, é necessário nesse caso alterar a versão do *kernel* instalado pelo *Fedora Core 1*.

A alteração necessária do *kernel*, será feita através da compilação de uma nova versão deste. No site [www.kernel.org](http://www.kernel.org), obtém-se os arquivos fontes do *kernel* necessário através do arquivo *linux-2.4.26.tar.bz2* .

### 3.8. Aplicação do *patch* e compilação do *kernel*

A descompactação do arquivo *linux-2.4.26.tar.bz2* no diretório `/usr/src`, é feita através do comando:

```
# tar xjvf linux-2.4.26.tar.bz2
```

Será criado o diretório `/usr/src/linux-2.4.26`, que contém a estrutura necessária para compilação do *kernel*.

A descompactação do arquivo *bruby-2.4.26-JD-20040626.diff.bz2* no

diretório `/usr/src`, é feita através do comando:

```
# bunzip2 bruby-2.4.26-JD-20040626.diff.bz2
```

Será disponibilizado o arquivo `/usr/src/bruby-2.4.26-JD-20040626.diff`, que é o *patch* a ser aplicado no *kernel*. Aplica-se o *patch* através do comando:

```
# patch -p -s0 < bruby-2.4.26-JD-20040626.diff
```

No diretório `/usr/src/linux-2.4.26` são configuradas as opções do *kernel* considerando os recursos desejados, exemplo de configuração que atenda essa implementação pode ser vista em <http://startx.times.lv/index.html>. Para configurar o *kernel*, utiliza-se o comando:

```
#make menuconfig
```

Após sair do “*make menuconfig*” salvando alterações, a compilação do *kernel* é feita através da execução dos comandos:

```
# make dep
# make modules
# make modules install
```

Após a compilação, deve-se configurar o ambiente para utilizar a nova versão de *kernel* obtida através dos comandos:

```
# rm -rf /lib/modules/2.4.26-b*
# make dep modules modules_install bzImage
# cp -f /usr/src/linux-2.4.26/arch/i386/boot/bzImage
```

```
/boot/vmlinuz-2.4.26-backstreet-ruby

# cp -f /usr/src/linux-2.4.26/System.map /boot/System.map-
2.4.26-backstreet-ruby

# cp -f /usr/src/linux-2.4.26/.config /boot/config-2.4.26-
backstreet-ruby

# cd /boot

# mkinitrd -f initrd-2.4.26-backstreet-ruby.img 2.4.26-
backstreet-ruby
```

Pode ser criado um pequeno *script* com os comandos relacionados anteriormente, objetivando agilizar a execução desses procedimentos. É de grande valia caso seja necessário executar várias compilações do *kernel* com objetivo de testar configurações variadas.

É necessário configurar o *GRUB* para disponibilizar a nova imagem do *kernel* como sendo uma das opções de *boot*. Principalmente na fase de testes, é importante manter a versão anterior do *kernel* disponível no *boot* para situações de emergência. Deve-se editar o arquivo de configuração do *GRUB* `/boot/grub/grub.conf` (figura 3.4) e inserir as linhas destacadas em **negrito**:

```
...
title Fedora Core (2.4.22-1.2115.nptl)
    root (hd0,1)
    kernel /vmlinuz-2.4.22-1.2115.nptl ro root=LABEL=/ rhgb
    initrd /initrd-2.4.22-1.2115.nptl.img

title Fedora Ruby
    root (hd0,1)
    kernel /vmlinuz-2.4.26-backstreet-ruby ro root=/dev/hda5 rhgb
devfs=mount dumbcon=3
    initrd /initrd-2.4.26-backstreet-ruby.img
```

Figura 3.4 – Conteúdo de `/boot/grub/grub.conf`

Observa-se que no item `root=/dev/hda5`, deve ser referenciada a partição relativa ao diretório “/”. Na penúltima linha do arquivo o identificador ***dumbcon=3*** indica a quantidade de terminais que serão utilizados com a contagem iniciando em 0, no caso 4 terminais.

### 3.9. Alteração de fonte de letra padrão do *Fedora Core 1*

Foi observado no primeiro *boot* executado com o novo *kernel* um travamento devido a fonte de letra padrão utilizada pelo *Fedora Core 1*, esse problema foi solucionado alterando-se a fonte padrão no arquivo `/etc/sysconfig/i18n`. Conteúdo original de `/etc/sysconfig/i18n` na figura 3.5, e conteúdo alterado de `/etc/sysconfig/i18n` na figura 3.6.

```
LANG="pt_BR.UTF-8  
SUPPORTED="pt_BR.UTF-8:pt_BR:pt"  
SYSFONT="latarcyrheb-sun16"
```

Figura 3.5 – Conteúdo original de `/etc/sysconfig/i18n`

```
LANG="pt_BR.UTF-8"  
SUPPORTED="pt_BR.UTF-8:pt_BR:pt"  
SYSFONT="lat1-16"
```

Figura 3.6 – Conteúdo alterado de `/etc/sysconfig/i18n`

### 3.10. Criação de dispositivos necessários para o *Multi-Terminal*

Como serão utilizados diversos conjuntos de *mouses* e teclados, deve-se

criar dispositivos que possibilitem o seu funcionamento. A criação dos dispositivos é feita executando os comandos:

```
# cd /dev
# mkdir input.old
# mv mouse js? input.old
# mkdir input
# cd input
# mknod js0 c 13 0
# mknod js1 c 13 1
# mknod js2 c 13 2
# mknod js3 c 13 3
# mknod mouse0 c 13 32
# mknod mouse1 c 13 33
# mknod mouse2 c 13 34
# mknod mouse3 c 13 35
# mknod mice c 13 63
# mknod event0 c 13 64
# mknod event1 c 13 65
# mknod event2 c 13 66
# mknod event3 c 13 67
# cd ..
# ln -s input/js0 js0
# ln -s input/js1 js1
# ln -s input/mice mouse
```

### 3.11. Instalação do pacote *devfsd*

O *devfs* faz com que o */dev* se comporte como um diretório virtual, assim como o */proc* e o */sys*, os *device nodes* são criados e destruídos de forma dinâmica, de acordo com o carregamento e descarregamento dos módulos. O pacote de instalação *devfsd-2.4.3-12.i386.rpm* pode ser obtido no site <http://www.rpmfind.net> e a instalação feita com o comando:

```
# rpm -ivh devfsd-2.4.3-12.i386.rpm
```

### 3.12. Instalação de *driver* para placas de vídeo

Apesar do *Fedora Core 1* disponibilizar um *driver* para a placa de vídeo NVIDIA modelo RIVA TNT2, notou-se que o mesmo não suporta o uso de várias placas de vídeo. Assim, obteve-se do *site* da NVIDIA ([www.nvidia.com](http://www.nvidia.com)) o *driver* da placa de vídeo para Linux *NVIDIA-Linux-x86-1.0-6111-pkg1.run* e instalar através do comando:

```
# sh NVIDIA-Linux-x86-1.0-6111-pkg1.run
```

Inserir-se no arquivo `/etc/rc.d/rc.local`, linha de comando para carregar o *driver* NVIDIA no *boot*:

```
# modprobe nvidia
```

### 3.13. *Boot* com o novo *kernel*

Após a instalação dos componentes do *hardware* e configurações iniciais, deve-se executar o *boot* com a nova imagem do *kernel* que foi gerada, observar que o único terminal que será ativado é o que foi denominado anteriormente de terminal principal.

### 3.14. Identificação do endereçamento de teclados e *mouses*

Os endereços dos teclados e *mouses* conectados ao Multi-Terminais, podem ser obtidos através da análise do arquivo `/proc/bus/input/devices`. Através dos endereços desses dispositivos, são definidos os conjuntos que irão atender cada um dos terminais. Exemplo parcial do arquivo `/proc/bus/input/devices` na figura 3.7:

```
I: Bus=0011 Vendor=0002 Product=0005 Version=0000
### Identificação do mouse PS/2
N: Name="ImPS/2 Generic Wheel Mouse"
P: Phys=isa0060/serio1/input0
H: Handlers=mouse0 event0
B: EV=7
B: KEY=70000 0 0 0 0 0 0 0 0
B: REL=103

I: Bus=0011 Vendor=0001 Product=0002 Version=ab02
### Identificação do teclado PS/2
N: Name="AT Set 2 keyboard"
P: Phys=isa0060/serio0/input0
H: Handlers=kbd event1
B: EV=120003
B: KEY=4 2000000 8061f9 fbc9d621 efdfffd ffeffff ffffffff
ffffffe
B: LED=7

I: Bus=0003 Vendor=0518 Product=0001 Version=0010
### Identificação do primeiro teclado USB
N: Name="CHESEN PS2 to USB Converter"
P: Phys=usb-00:10.0-1/input0
H: Handlers=kbd event2
B: EV=12000b
B: KEY=10000 7f ffe00000 7ff ffbefdfd ffffffff ffffffff ffffffe
B: ABS=100 0
B: LED=1f07
```

Figura 3.7 – Conteúdo de `/proc/bus/input/devices`



Através do conteúdo deste arquivo, obtém-se informações importantes através dos identificadores:

- **Name**= Fornece o nome do dispositivo
- **Phys**= Fornece o endereçamento do dispositivo
- **Handlers**= Especifica se o dispositivo existente no endereço é teclado ou *mouse*.

Os endereços definidos através do identificador **Phys**, são utilizados na construção dos arquivos `/etc/hotplug/mouse.conf` e `/etc/hotplug/kbd.conf`, que serão tratados a seguir.

### 3.15. Utilizando *hotplug* com *input.agent* e *input.rc*

O *hotplug* é uma ferramenta de detecção e instalação automática de novos dispositivos para *GNU/Linux*. Quando executado, o *hotplug* realiza uma série de testes que detecta qualquer novo dispositivo de *hardware* e automaticamente carrega o módulo adequado, se disponível. É comum o *hotplug* vir configurado para ser executado toda vez que o sistema for iniciado, assim qualquer novo *hardware* que seja conectado ao computador, será detectado e instalado durante o processo de inicialização. Outra função do *hotplug* é detectar dispositivos USB quando eles são conectados ao computador, instalando automaticamente os módulos necessários.

Utilizando-se os *scripts* *input.agent* (apêndice B) e *input.rc* (apêndice C), gravados no diretório `/etc/hotplug`, pode-se gerenciar os dispositivos de entrada através de seus endereços fixos que são obtidos através de `/proc/bus/input/devices`. O *script* *input.rc* executará *input.agent* com os

argumentos apropriados de drivers que foram compilados no *kernel* ou que foram carregados antes que o *hotplug* esteja disponível. O arquivo *input.agent*, utiliza como argumentos os conteúdos dos arquivos `/etc/hotplug/mouse.conf` (figura 3.8) e `/etc/hotplug/kbd.conf` (figura 3.9) que possuem como conteúdo os endereços dos dispositivos de *mouse* e teclados conectados a *CPU*. O arquivo *input.agent* pode ser obtido em [www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/scripts\\_input-rc.html](http://www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/scripts_input-rc.html) e o arquivo *input-agent* em [http://www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/scripts\\_input-agent.html](http://www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/scripts_input-agent.html). Deve-se dar permissão de execução aos arquivos após serem gravados em `/etc/hotplug`.

```
# mouse configuration
#
# sym_link      device_physicaly_location
mouse0         isa0060/serio1/input0
mouse1         usb-00:10.0-1/input1
mouse2         usb-00:10.0-2/input1
mouse3         usb-00:10.1-1/input1
```

Figura 3.8 – Conteúdo do arquivo `/etc/hotplug/mouse.conf`

```
# keyboard configuration
#
# vt_name       device_physicaly_location
VT0            isa0060/serio0/input0
VT1            usb-00:10.0-1/input0
VT2            usb-00:10.0-2/input0
VT3            usb-00:10-1-1/input0
```

Figura 3.9 – Conteúdo do arquivo `/etc/hotplug/kbd.conf`

### 3.16. O serviço *ruby\_init*

Deve ser gravado no diretório `/etc/rc.d/init.d` o *script* *ruby\_init* (apêndice D) que tem como função configurar/ativar serviços específicos do

*Backstreet Ruby* como carregar módulos adicionais de entrada, gerenciar o *Xfree* e arquivos de configuração de vídeo. Esse *script* pode ser obtido em [tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/app\\_ruby\\_init\\_service.html](http://tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/app_ruby_init_service.html), deve-se dar permissão de execução ao arquivo após ser gravado em `/etc/rc.d/init.d` e deve ser executado automaticamente quando o sistema for ligado, para que isso execute-se o comando:

```
# chkconfig --add ruby_init
```

O `script` `ruby_init` quando executado, lê o arquivo de configuração `/etc/sysinit/ruby.conf` (figura 3.10). Este arquivo é obtido em [tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/app\\_ruby\\_init\\_ruby\\_conf.html](http://tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/app_ruby_init_ruby_conf.html) e deve ter seu conteúdo alterado de forma a identificar alias e endereços dos dispositivos de teclado e *mouse* de acordo com o que foi definido nos arquivos `/etc/hotplug/kbd.conf` e `/etc/hotplug/mouse.conf`. As alterações em questão estão destacadas em **negrito**.

```

...
# keyboard devices configuration
#
# vt_name device_physicaly_location

VT0      isa0060/serio0/input0
VT1      usb-00:10.0-1/input0
VT2      usb-00:10.0-2/input0
VT3      usb-00:10.1-1/input0

...

#
# mouse devices configuration
#
# sym_link device_physicaly_location

mouse0    isa0060/serio1/input0
mouse1    usb-00:10.0-1/input1
mouse2    usb-00:10.0-2/input1
mouse3    usb-00:10.1-1/input1

...

#
# event devices configuration
#
# sym_link device_physicaly_location
#event0br    isa0060/serio1/input0
#event1br    usb-00:10.1-1.1/input0

# end input device configuration
#

```

Figura 3.10 – Conteúdo do arquivo /etc/sysconfig/ruby.conf

### 3.17. Configuração do serviço de vídeo

O serviço de vídeo deve ser alterado de forma que possa atender a todos os terminais que compõem o sistema. O primeiro passo é instalar um servidor X que possibilite atender diversas placas de vídeo. Deve ser obtido no site

<http://startx.times.lv/> o arquivo *Xfree86-430-prefbusid3.bz2* que é um servidor X que cumpre esse papel. Esse arquivo deve ser descompactado através do comando:

```
# bzunzip2 Xfree86-430-prefbusid3.bz2
```

O arquivo *Xfree86-430-prefbusid3* obtido com a descompactação, deve ser copiado para o diretório `/usr/X11R6/bin` com o comando:

```
# cp Xfree86-430-prefbusid3 /usr/X11R6/bin
```

Para facilitar a configuração e até mesmo a documentação dos arquivos de configuração a serem utilizados, é interessante criar um *link* de *Xfree86-450-prefbusid3* para ser utilizado com cada um dos terminais instalados:

```
# ln -s /usr/X11R6/bin/Xfree86-430-prefbusid3 /usr/X11R6/bin/X1
# ln -s /usr/X11R6/bin/Xfree86-430-prefbusid3 /usr/X11R6/bin/X2
# ln -s /usr/X11R6/bin/Xfree86-430-prefbusid3 /usr/X11R6/bin/X3
# ln -s /usr/X11R6/bin/Xfree86-430-prefbusid3 /usr/X11R6/bin/X4
```

Deve ser criado um arquivo de configuração de vídeo no diretório `/etc/X11` nos padrões do *XF86Config* para cada uma dos terminais utilizados. Esses arquivos na seção “*Device*” devem identificar o driver de vídeo em uso (*nvidia*) e o endereço da placa de vídeo. Os endereços das placas de vídeo são obtidas através da execução do comando:

```
# lspci | grep VGA
```

Exemplo de endereços resultantes são mostradas em negrito na figura 3.11.

```
00:0a.0 VGA compatible controller: nVidia Corporation NV5M64
[RIVA TNT2 Model 64/Model 64 Pro] (rev 15)
00:0d.0 VGA compatible controller: nVidia Corporation NV5M64
[RIVA TNT2 Model 64/Model 64 Pro] (rev 15)
00:13.0 VGA compatible controller: nVidia Corporation NV5M64
[RIVA TNT2 Model 64/Model 64 Pro] (rev 15)
01:00.0 VGA compatible controller: nVidia Corporation NV5M64
[RIVA TNT2 Model 64/Model 64 Pro] (rev 15)
```

Figura 3.11 – Endereços das placas de vídeo

Nos arquivos de configuração a seção “*ServerLayout*” deve indicar os *links* (X1, X2, X3 ou X4), referentes ao servidor de vídeo *Xfree86-450-prefbusid3* . Nas seções “*InputDevice*” devem ser identificados teclados e *mouses* e na seção “*Monitor*” as características do monitor utilizado em conjunto com cada placa de vídeo. Exemplo de parte de um arquivo configurado (/etc/X11/XFreeConfig1) é apresentado na figura 3.12, deve ser criado um arquivo similar para cada terminal em uso. As alterações deste arquivo estão em **negrito**.



arquivo `/etc/X11/gdm/gdm.conf`, configurando a quantidade e quais os terminais que devem ser atendidos. Nas figuras 3.13a, 3.13b e 3.13c são apresentadas partes do conteúdo de um arquivo `/etc/X11/gdm/gdm.conf` alterado para gerenciar o *login* em quatro terminais. Em negrito estão destacados as alterações efetuadas no arquivo original.

```
...
User=gdm
Group=gdm
# To try to kill all clients started at greeter time or in the
Init script.
# doesn't always work, only if those clients have a window of
their own

#KillInitClients=true    >>> Valor Original
KillInitClients=true

LogDir=/var/log/gdm
# You should probably never change this value unless you have a
weird setup
PidFile=/var/run/gdm.pid

...

DisplayInitDir=/etc/X11/gdm/Init
# Distributions:  If you have some script that runs an X server
in say
# VGA mode, allowing a login, could you please send it to me?

#FailsafeXServer=    >>> Valor Original
FailsafeXServer=True

# if X keeps crashing on us we run this script.  The default one
does a bunch
# of cool stuff to figure out what to tell the user and such and
can
# run an X configuration program.

...
```

Figura 3.13a – Primeira parte do conteúdo do arquivo `/etc/X11/gdm/gdm.conf`



```

#StandardXServer=/usr/X11R6/bin/X      >>> VALOR ORIGINAL
StandardXServer=/usr/X11R6/bin/X1

# The maximum number of flexible X servers to run.

#FlexibleXServers=5      >>> VALOR ORIGINAL
FlexibleXServers=5

# the X nest command
Xnest=/usr/X11R6/bin/Xnest -audit 0 -name Xnest
# Automatic VT allocation.  Right now only works on Linux.  This
way

#FirstVT=7      >>> VALOR ORIGINAL
#VTAllocation=true >>> VALOR ORIGINAL

FirstVT=7
VTAllocation=true

# Should double login be treated with a warning (and possibility
to change
# vts on linux systems for console logins)
#DoubleLoginWarning=true

...

[servers]
# These are the standard servers.  You can add as many you want
here
# and they will always be started.  Each line must start with a
unique
# number and that will be the display number of that server.
Usually just
# the 0 server is used.

0=Standard
### INSERIDOS
1=2nd
2=3nd
3=4nd

# Note the VTAllocation and FirstVT keys on linux.  Don't add any
vt<number>

...

```

Figura 3.13b – Segunda parte do conteúdo do arquivo /etc/X11/gdm/gdm.conf

```

# Definition of the standard X server.
[server-Standard]
name=Standard server
### command=/usr/X11R6/bin/X :0 -xf86config /etc/X11/XF86Config
-deferglyphs 16 -ac vt7 >>> ORIGINAL
command=/usr/X11R6/bin/X0 :0 -xf86config /etc/X11/XF86Config-0
-deferglyphs 16 -ac vt7
flexible=true

## INSERIDAS REFERÊNCIA PARA OS NOVOS TERMINAIS
[server-2nd]
name=2nd server
command=/usr/X11R6/bin/X1 :1 -xf86config /etc/X11/XF86Config-1
-deferglyphs 16 vt17
flexible=true

[server-3nd]
name=3nd server
command=/usr/X11R6/bin/X2 :2 -xf86config /etc/X11/XF86Config-2
-deferglyphs 16 vt18
flexible=true

[server-4nd]
name=4nd server
command=/usr/X11R6/bin/X3 :3 -xf86config /etc/X11/XF86Config-3
-deferglyphs 16 vt19
flexible=true

# To use this server type you should add -query host or -indirect
host
# to the command line
[server-Terminal]

...

```

Figura 3.13c – Terceira parte do conteúdo do arquivo /etc/X11/gdm/gdm.conf

### 3.18. Testes iniciais

Após efetuadas as configurações, reinicia-se o computador e observa-se a sequência de ativação dos terminais. O terminal ligado a placa de vídeo AGP que no *setup* foi configurada para ser a principal no sistema e é o primeiro a ser ativado. Nas figura 3.14, são mostrados os terminais ativos apresentando as telas

de login.



Figura 3.14 – Terminais apresentando tela de login

Para efeitos de teste, efetua-se *login* como usuário *root* e cadastra-se usuários e senhas. Deve-se efetuar *login* com os diferentes usuários nos demais terminais e executam-se programas diversos conforme apresentado na figura 3.15.



Figura 3.15 – Terminais em uso

Na figura 3.16, é apresentada parte da listagem dos processos referentes aos terminais e diversos usuários (destacados em negrito) .

USER	PID	STAT	START	COMMAND
root	1	S	13:19	init
root	2	SW	13:19	[keventd]
...				
mysql	3224	S	13:19	/usr/libexec/mysqld
mysql	3225	S	13:19	/usr/libexec/mysqld
...				
root	3378	S	13:19	/usr/bin/gdm-binary
root	3407	S	13:19	xdm
<b>root</b>	<b>3422</b>	<b>S</b>	<b>13:19</b>	<b>/usr/bin/gdm-binary</b>
<b>root</b>	<b>3423</b>	<b>SL</b>	<b>13:19</b>	<b>/usr/X11R6/bin/X0</b>
<b>root</b>	<b>3429</b>	<b>S</b>	<b>13:20</b>	<b>/usr/bin/gdm-binary</b>
<b>root</b>	<b>3436</b>	<b>SL</b>	<b>13:20</b>	<b>/usr/X11R6/bin/X1</b>
<b>root</b>	<b>3440</b>	<b>S</b>	<b>13:20</b>	<b>/usr/bin/gdm-binary</b>
...				
<b>547</b>	<b>4075</b>	<b>R</b>	<b>13:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>547</b>	<b>4077</b>	<b>S</b>	<b>13:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>547</b>	<b>4078</b>	<b>S</b>	<b>13:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
...				
<b>549</b>	<b>4692</b>	<b>S</b>	<b>13:42</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>549</b>	<b>4694</b>	<b>S</b>	<b>13:42</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
...				
547	4712	S	13:44	bluefish
543	4771	S	13:52	bluefish
...				
<b>dionerf</b>	<b>5015</b>	<b>S</b>	<b>13:55</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>dionerf</b>	<b>5017</b>	<b>S</b>	<b>13:55</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
...				
glauberb	5800	S	14:10	/usr/lib/firefox-1.0.4/firefox-bin
...				
<b>543</b>	<b>13386</b>	<b>S</b>	<b>15:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>543</b>	<b>13387</b>	<b>S</b>	<b>15:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
<b>543</b>	<b>13389</b>	<b>S</b>	<b>15:39</b>	<b>/usr/lib/firefox-1.0.4/firefox-bin</b>
root	13601	S	16:31	sshd:
...				

Figura 3.16 – Processos referentes aos terminais e diferentes usuários

Caso ocorram problemas que impeçam o *boot* e utilização dos terminais, executa-se um *reboot* na máquina e seleciona-se no *GRUB* o *boot* pelo *kernel* original que foi mantido. Assim, terá acesso ao sistema e poderá iniciar a localização dos problemas.

## 4. ANÁLISE DE CUSTOS

A implementação do Multi-Terminais, implica em custos adicionais de *hardware* se comparado com um microcomputador com configuração tradicional. Na tabela 4.1<sup>1</sup> são apresentados esses valores considerando 3 unidades a mais de cada um dos itens que são necessários para a implementação de 4 terminais, exceto para memória *RAM* (somente 1 item adicional).

<b>Itens adicionais</b>	<b>Valor Unitário</b>	<b>Valor Total</b>
Placa de vídeo TNT2 PCI	R\$ 129,00	R\$ 387,00
Mouse	R\$ 18,00	R\$ 54,00
Teclado	R\$ 24,00	R\$ 72,00
Conversor USB/PS-2	R\$ 34,00	R\$ 102,00
Monitor de vídeo	R\$ 349,00	R\$ 1047,00
	<b>Total Geral</b>	<b>R\$ 1662,00</b>

Tabela 4.1 – Preços de itens de *hardware* adicionais

Para efeitos de análise e comparação de custos, serão tomados como base os microcomputadores modelo Dell Dimension 1100n com monitor de vídeo, memória *RAM* de 256MB, sem sistema operacional instalado e o modelo Dell OptiPlex com memória *RAM* de 1GB. Na tabela 4.2<sup>2</sup> é demonstrado o valor a ser gasto na aquisição de 4 unidades do Dell Dimension 1100n.

1 Fonte: HardSoft – Bauru/SP em 21/07/06

2 Fonte: [www.dell.com.br](http://www.dell.com.br) em 21/07/06

<b>Equipamento</b>	<b>Valor unitário</b>	<b>Valor de 4 unidades</b>
Dell Dimension 1100n	R\$ 1301,00	<b>R\$ 5204,00</b>

Tabela 4.2 – Gasto na aquisição de microcomputadores

Na tabela 4.3, é apresentada a planilha de gastos com a implementação de um sistema com 4 terminais utilizando a *CPU* modelo Dell OptiPlex.

<b>Equipamentos</b>	<b>Valor unitário</b>
Dell OptiPlex GX620	R\$ 2396,00
<i>Hardware</i> adicional	R\$ 1812,00
<b>Valor Total</b>	<b>R\$ 4208,00</b>

Tabela 4.3 – Gastos na montagem de *Multi-Terminais*

Mesmo utilizando uma *CPU* com maior capacidade de processamento, o custo de implementação do Multi-Terminal nesta comparação foi 19,1% menor do que uma configuração padrão de 4 desktops.

## 5. CONCLUSÃO

O uso de Multi Terminais *GNU/Linux*, é uma alternativa extremamente interessante na montagem de laboratórios e telecentros, pois representa economia no momento da aquisição de *hardware* e sua implementação. Além dessa economia inicial, devem ser considerados também os seguintes pontos positivos dessa solução:

- Menor consumo de energia elétrica, devido a quantidade menor de *CPUs*.
- Menor aquecimento do ambiente.
- Economia na compra de estabilizadores/*no-break*.
- Economia na aquisição e instalação de cabeamento e equipamentos de rede (1 ponto de rede atende todos os terminais).
- Necessidade de menor espaço físico para instalação e mobiliário.
- É ecologicamente correto pois resulta em menos material de descarte (lixo), quando os equipamentos forem desativados.

A possibilidade de uso de uma variedade de outros *software* gratuito, é com certeza outro fator que deve ser considerado pois representa economia para a entidade que optar por essa solução e também significa ter liberdade para escolher.



Como em todo sistema onde se tem centralização de processamento, o Multi-Terminais tem como ponto fraco a possibilidade de falha ou queima da *CPU*. A ocorrência de tal fato, deixaria todos os usuários sem sua ferramenta de trabalho.

Mesmo sendo utilizado na execução de programas que exigem grande processamento por parte da *CPU*, o sistema de Multi-Terminais apresenta uma performance que torna o seu uso bastante produtivo para os usuários.

Outras técnicas para implementação de Multi-Terminais *GNU/Linux* surgiram depois do *patch Backstreet Ruby* que foi o alvo deste trabalho. "*Fica como proposta para futuros trabalhos a implementação utilizando novas técnicas, como por exemplo Xephyr*".<sup>1</sup>

---

<sup>1</sup> *Xephyr* – [www.c3sl.ufpr.br/multiterminal/howtos/howto-xephyr-pt.htm](http://www.c3sl.ufpr.br/multiterminal/howtos/howto-xephyr-pt.htm)

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

C3SL – Centro de computação e software livre da UFPR – disponível em <http://www.c3sl.ufpr.br/multiterminal/index-pt.php> - Acessado em 22/04/2006

FILHO, Antonio Mendes da Silva – *Artigo Os três pilares da inclusão digital* – Publicado na Revista Espaço Acadêmico – Ano III – N° 24 – Maio de 2003 – ISSN 1519.6186 -

Disponível em <http://www.espacoacademico.com.br/024/24amsf.htm> – Acessado em 12/01/2006

FREITAS, Miguel - *Multiple local XFree users under Linux* – Disponível em <http://cambuca.ldhs.cetuc.puc-rio.br/multiuser/> - Acessado em 29/05/2006

IWS – *Internet World Stats – Estatística de uso da Internet por Regiões do Mundo em 31/03/2006* – Disponível em <http://www.internetworldstats.com/stats.htm> – Acessado em 03/05/2006

LITTO, Fredric Michael – *Artigo Telecentros comunitários - Uma resposta à "exclusão digital"* – Disponível em <http://www.lidec.futuro.usp.br/artigo3.php> – Acessado em 22/03/2006

MACADAR, Marie Anne – *Artigo Desmistificando a inclusão digital* – Publicado em *Integração - A revista eletrônica do Terceiro Setor* - Disponível em <http://integracao.fgvsp.br/ano5/20/opiniaio.htm> – Acessado em 20/01/2006

MULTITERMINAL – *Artigo Um modelo de computação multiusuário baseado*

*em microcomputadores pessoais* – Disponível em <http://web.inf.ufpr.br/tv02/papers/artigo-fisl.pdf> – Acessado em 20/05/2006

PORTAL – *Software livre na Prefeitura de São Paulo* – Disponível em [http://PORTAL.prefeitura.sp.gov.br/cidadania/coordenadoria\\_governo\\_eletronico/software\\_livre/0004](http://PORTAL.prefeitura.sp.gov.br/cidadania/coordenadoria_governo_eletronico/software_livre/0004) – Acessado em 14/03/2006

REBÊLO, Paulo – *Artigo Inclusão digital: o que é e a quem se destina ?* - Disponível em <http://webinsider.uol.com.br/vernoticia.php/id/2443> – Acessado em 15/04/2006

RUBY, *Site oficial do Backstreet Ruby* – Disponível em [http://www.ltn.lv/~aivils/?proj\\_id=ruby](http://www.ltn.lv/~aivils/?proj_id=ruby) – Acessado em 22/04/2006

SABACK, Wagner e Giselia Cruz - *Artigo Inclusão Digital – com cepções e Ações em Jogo* – Disponível em <http://twiki.im.ufba.br/bin/view/PSL/OQueEIclusaoDigital> – Acessado em 30/04/2006

SEABRA, Carlos – *Artigo Inclusão digital: desafios maiores que as simples boas intenções* – Disponível em <http://www.lidec.futuro.usp.br/artigo%20SEABRA.php> – Acessado em 03/05/2006

X, *Definição de X Window System* – Disponível em [http://pt.wikipedia.org/wiki/X\\_Window\\_System](http://pt.wikipedia.org/wiki/X_Window_System) – Acessado em 22/04/2006

XFree Local Multi-User HOWTO – Autor: Author: Svetoslav Slavtchev –  
disponível em <http://www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/>  
- Acessado em 22/04/2006

## APÊNDICES

### Apêndice A - Listagem do arquivo /boot/grub/grub.conf

```
# Note that you do not have to rerun grub after making changes to
this file
# NOTICE: You have a /boot partition. This means that
#           all kernel and initrd paths are relative to /boot/,
eg.
#           root (hd0,1)
#           kernel /vmlinuz-version ro root=/dev/hda5
#           initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=10
splashimage=(hd0,1)/grub/splash.xpm.gz
title Fedora Core (2.4.22-1.2115.nptl)
    root (hd0,1)
    kernel /vmlinuz-2.4.22-1.2115.nptl ro root=LABEL=/ rhgb
    initrd /initrd-2.4.22-1.2115.nptl.img

title Fedora Ruby
    root(hd0,1)
    kernel /vmlinuz-2.4.26-backstreet-ruby ro root=/dev/hda5 rhgb
devfs=mount dumbcon=3
    initrd /initrd-2.4.26-backstreet-ruby.img
```

## Apêndice B - Listagem do arquivo / proc/bus/input/devices

```
I: Bus=0011 Vendor=0002 Product=0005 Version=0000

###   Identificação do mouse PS/2
N: Name="ImPS/2 Generic Wheel Mouse"
P: Phys=isa0060/serio1/input0
H: Handlers=mouse0 event0
B: EV=7
B: KEY=70000 0 0 0 0 0 0 0 0
B: REL=103

I: Bus=0011 Vendor=0001 Product=0002 Version=ab02

###   Identificação do teclado PS/2
N: Name="AT Set 2 keyboard"
P: Phys=isa0060/serio0/input0
H: Handlers=kbd event1
B: EV=120003
B: KEY=4 2000000 8061f9 fbc9d621 efdfffd ffeffff ffffffff
ffffffe
B: LED=7

I: Bus=0003 Vendor=0518 Product=0001 Version=0010

###   Identificação do primeiro teclado USB
N: Name="CHESEN PS2 to USB Converter"
P: Phys=usb-00:10.0-1/input0
H: Handlers=kbd event2
B: EV=12000b
B: KEY=10000 7f ffe00000 7ff ffbefddf ffffffff ffffffff ffffffff
B: ABS=100 0
B: LED=1f07

I: Bus=0003 Vendor=0518 Product=0001 Version=0010

###   Identificação do primeiro mouse USB
N: Name="CHESEN PS2 to USB Converter"
P: Phys=usb-00:10.0-1/input1
H: Handlers=kbd mouse1 event3
B: EV=f
B: KEY=ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff f
fff0000 0 397a d843d7a9 1e0000 0 0 1
B: REL=103
B: ABS=100 0

I: Bus=0003 Vendor=0d3d Product=0001 Version=0001
```

```

### Identificação do segundo teclado USB
N: Name="0d3d:0001"
P: Phys=usb-00:10.0-2/input0
H: Handlers=kbd event4
B: EV=12000b
B: KEY=10000 7f ffe00000 7ff ffbefdf ffffffff ffffffff fffffffe
B: ABS=300 0
B: LED=1f

I: Bus=0003 Vendor=0d3d Product=0001 Version=0001

### Identificação do segundo mouse USB
N: Name="0d3d:0001"
P: Phys=usb-00:10.0-2/input1
H: Handlers=kbd mouse2 event5
B: EV=f
B: KEY=ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffff0000 0 397a d843d7a9 1e0000 0 0 1
B: REL=103
B: ABS=100 0

I: Bus=0003 Vendor=0d3d Product=0001 Version=0001

### Identificação do terceiro teclado USB
N: Name="0d3d:0001"
P: Phys=usb-00:10.1-1/input0
H: Handlers=kbd event6
B: EV=12000b
B: KEY=10000 7f ffe00000 7ff ffbefdf ffffffff ffffffff fffffffe
B: ABS=300 0
B: LED=1f

I: Bus=0003 Vendor=0d3d Product=0001 Version=0001

### Identificação do terceiro mouse USB
N: Name="0d3d:0001"
P: Phys=usb-00:10.1-1/input1
H: Handlers=kbd mouse3 event7
B: EV=f
B: KEY=ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffff0000 0 397a d843d7a9 1e0000 0 0 1
B: REL=103
B: ABS=100 0

```

## Apêndice C - Listagem do arquivo /etc/hotplug/input.agent

```
#!/bin/sh
# Please place this file /etc/hotplug
#
# input-specific hotplug policy agent.
#
# Kernel Input params are:
#
#     ACTION=add
#     PHYS=physical location of device
#     NAME=Name of the device
#
# HISTORY:
#     15-JUN-2003     removed paste
#     07-MAY-2003     remake by Aivils Stoss
#                     /proc manipulation added
#                     parse kbd.conf event.conf mouse.conf.
#                     create necessary symbolic links
#     28-SEP-2002     Initial version from Andreas Schuldei
#                     andreas (at) schuldei.org
#
cd /etc/hotplug
. hotplug.functions
DEBUG=yes export DEBUG

KBD_CONFIG="./kbd.conf"
EVENT_CONFIG="./event.conf"
MOUSE_CONFIG="./mouse.conf"

setup_kbd ()
{
    while read VT_NUM PHYS_PATTERN NAME_PATTERN
    do
        if [ `echo "$VT_NUM" | grep "^#"` ]; then
            continue;
        fi
        if [ `echo "$PHYS" | grep $PHYS_PATTERN 2>/dev/null` ];
    then
        VT=`echo "$VT_NUM" | sed 's/VT//' | awk '{printf
"%02d", $0}`
        if [ -d /proc/bus/console/$VT ]; then
            echo "$PHYS" > "/proc/bus/console/$VT/keyboard"
            debug_mesg "Input device $NAME on $PHYS mapping
as secondary to VT$VT"
            return;
        else
            debug_mesg "Trying to configure keyboard for VT$VT ,
```



```

but not enough VT's available"
    fi
done
debug_mesg "Found no fitting VT"
}

setup_event ()
{
    while read SYM_LINK PHYS_PATTERN NAME_PATTERN
    do
        if [ `echo "$SYM_LINK" | grep "^#"` ]; then
            continue;
        fi
        if [ `echo "$PHYS" | grep $PHYS_PATTERN 2>/dev/null` ];
then
            case $ACTION in
            add)
                cd /dev/input
                rm -f $SYM_LINK
                ln -s $DEV_EVENT $SYM_LINK
                debug_mesg "Input event device $NAME on $PHYS linked
to $SYM_LINK"
                ;;
            remove)
                rm -f /dev/input/$SYM_LINK
                debug_mesg "Input event device link $SYM_LINK
removed"
                ;;
            esac
            return;
        fi
    done
    debug_mesg "Found no fitting event device"
}

setup_mouse ()
{
    while read SYM_LINK PHYS_PATTERN NAME_PATTERN
    do
        if [ `echo "$SYM_LINK" | grep "^#"` ]; then
            continue;
        fi
        if [ `echo "$PHYS" | grep $PHYS_PATTERN 2>/dev/null` ];
then
            case $ACTION in
            add)
                cd /dev/input
                rm -f $SYM_LINK

```

```

        ln -s $DEV_MOUSE $SYM_LINK
        debug_mesg "Input mouse device $NAME on $PHYS linked
to $SYM_LINK"
        ;;
        remove)
        rm -f /dev/input/$SYM_LINK
        debug_mesg "Input mouse device link $SYM_LINK
removed"
        ;;
    esac
    return;
fi
done
debug_mesg "Found no fitting mouse device"
}

if [ "$ACTION" = "" ]; then
    mesg Bad input agent invocation
    exit 1
fi

DEV_HANDLERS=`grep -E 'Phys|Handlers' /proc/bus/input/devices | \
awk '{ if(count == 0) { printf("%s\t", $0); count++; } else
{ print $0; count=0; } }' | \
grep $PHYS | awk -F\t '{print $2}' | sed 's/^\.*=//'\`

# older grep do not support -o :- (
#DEV_EVENT=`echo $MATCHED | grep -oE event[0-9]+`
#DEV_MOUSE=`echo $MATCHED | grep -oE mouse[0-9]+`
#DEV_KBD=`echo $MATCHED | grep -o kbd`

DEV_EVENT=`echo $DEV_HANDLERS | \
awk -F" " '{for(n=1;$n;n=n+1) if($n ~ /event/) print
$n}'`
DEV_MOUSE=`echo $DEV_HANDLERS | \
awk -F" " '{for(n=1;$n;n=n+1) if($n ~ /mouse/) print
$n}'`
DEV_KBD=`echo $DEV_HANDLERS | \
awk -F" " '{for(n=1;$n;n=n+1) if($n ~ /kbd/) print
$n}'`

#
# What to do with this input device event?
#
case "$ACTION" in

add)
    if [ -n "$DEV_KBD" ]; then
        setup_kbd < $KBD_CONFIG
    fi
fi

```

```
fi
if [ -n "$DEV_EVENT" ]; then
    setup_event < $EVENT_CONFIG
fi
if [ -n "$DEV_MOUSE" ]; then
    setup_mouse < $MOUSE_CONFIG
fi
;;
remove)
    #setup_event < $EVENT_CONFIG
    #setup_mouse < $MOUSE_CONFIG
    ;;
*)
    debug_mesg "Input '$ACTION' event not supported"
    return 1
    ;;
esac
```

## Apêndice D - Listagem do arquivo /etc/hotplug/input.rc

```
#!/bin/bash
#
# input.rc This loads handlers for those input devices
#          that have drivers compiled in kernel
#          Currently stopping is not supported
#
# Best invoked via /etc/init.d/hotplug or equivalent, with
# writable /tmp, /usr mounted, and syslogging active.
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin
PROCDIR=/proc/bus/input

# source function library
if [ -f /etc/init.d/functions ]; then
    . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ]; then
    . /etc/rc.d/init.d/functions
fi

if [ -f /etc/hotplug/hotplug.functions ]; then
    . /etc/hotplug/hotplug.functions
fi

input_reset_state () {
    PRODUCT=
    NAME=
    PHYS=
    EV=
    KEY=
    REL=
    ABS=
    MSC=
    LED=
    SND=
    FF=
}

#
# "COLD PLUG" ... load input handlers for compile-in input
drivers loaded
# before the OS could really handle hotplug, perhaps because
/sbin or
```

```

# $HOTPLUG_DIR wasn't available or /tmp wasn't writable. When/if
the
# /sbin/hotplug program is invoked then, hotplug event
notifications
# get dropped. To make up for such "cold boot" errors, we
synthesize
# all the hotplug events we expect to have seen already. They
can be
# out of order, and some might be duplicates.
#
input_boot_events ()
{
    if [ ! -r $PROCDIR/devices ]; then
        echo $"*** can't synthesize input events -
$PROCDIR/devices missing"
        return
    fi

    ACTION=add
    export ACTION

    export PRODUCT NAME PHYS EV KEY REL ABS MSC LED SND FF
    input_reset_state

    declare line

    #
    # the following reads from /proc/bus/input/devices. It is
inherently
    # racy (esp. as this file may be changed by input.agent
invocation)
    # but so far input devices do not appear in sysfs
    #
    while read line; do
        case "$line" in
            I:* ) # product ID
                eval "${line#I: }"
                PRODUCT=$Bus/$Vendor/$Product/$Version
                ;;
            N:* ) # name
                eval "${line#N: }"
                NAME="$Name"
                ;;
            P:* ) # Physical
                eval "${line#P: }"
                PHYS=$Phys
                ;;
            B:* ) # Controls supported
                line="${line#B: }"

```

```

eval "${line%%=*}=\"${line#*=}\""
;;
"" )      # End of block
debug_mesg "Invoking input.agent"
debug_mesg "PRODUCT=$PRODUCT"
debug_mesg "NAME=$NAME"
debug_mesg "PHYS=$PHYS"
debug_mesg "EV=$EV"
debug_mesg "KEY=$KEY"
debug_mesg "REL=$REL"
debug_mesg "ABS=$ABS"
debug_mesg "MSC=$MSC"
debug_mesg "LED=$LED"
debug_mesg "SND=$SND"
debug_mesg "FF=$FF"
/etc/hotplug/input.agent < /dev/null
input_reset_state
;;
esac
done < $PROCDIR/devices
}

# See how we were called.
case "$1" in
start)
input_boot_events
;;
stop)
: not supported currently
;;
status)
echo "$INPUT status for kernel: " `uname -srm`
echo ''

echo "INPUT devices:"
if [ -r $PROCDIR/devices ]; then
grep "^[INHP]:" $PROCDIR/devices
else
echo "$PROCDIR/devices not available"
fi
echo ''

echo "INPUT handlers:"
if [ -r $PROCDIR/handlers ]; then
cat $PROCDIR/handlers
else
echo "$PROCDIR/handlers not available"
fi

```

```
    echo ''
    ;;
restart)
    # always invoke by absolute path, else PATH=$PATH:
    $0 stop && $0 start
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart}"
    exit 1
esac
```

## Apêndice E - Listagem do arquivo /etc/rc.d/init.d/ruby\_init

```
#!/bin/sh
#
# ruby_init This scripts configures cold-plugging for bruby, \
#           loads additional input modules, manages the XFree \
#           and display manager configuration files
#
# chkconfig: 2345 02 98
# description:    Configures the Bruby input subsystem and
manages \
#               XFree and display manager configuration files.
# config: /etc/sysconfig/ruby.conf
#
# TODO
#     * handle commented out "PciOsConfig"
#     *? status/ ext-status
#     - show config
#     - check wether curr. configuration matches setup ?
#     - make it work when hotplug files not installed ?
#     ( currently it will just just inform that they are not
installed
#     and exit )
#
# $Id: ruby_init,v 0.0.5 2003/10/12 12:32 svetljo Exp $
# - don't exit silently if hackvideo or modules loading is
deactivated,
#   run the script to the end
# $Id: ruby_init,v 0.0.4 2003/09/13 18:42 svetljo Exp $
# - show keyboard status per VT ,use sed instead of head(head is
in /usr)
# - *? status/ ext-status
# - (mice & evdev links)?
# $Id: ruby_init,v 0.0.3 2003/09/13 11:40 svetljo Exp $
# - include hackvideo handling (long time wondering what was
missing)
# $Id: ruby_init,v 0.0.2 2003/09/12 18:12 svetljo Exp $
# - got it actually running
# - use >& /dev/null to load variables from the config file
#   (what a mess)
# $Id: ruby_init,v 0.0.1 2003/09/12 12:01 svetljo Exp $
# - initial release

PATH=/sbin:/bin:/usr/sbin:/usr/bin

# source function library
if [ -f /etc/init.d/functions ]; then
. /etc/init.d/functions
```



```

elif [ -f /etc/rc.d/init.d/functions ]; then
    . /etc/rc.d/init.d/functions
fi

# source defaults
if [ -f /etc/sysconfig/ruby.conf ]; then
    #WTF, how to get rid of the "... >& /dev/null"
    . /etc/sysconfig/ruby.conf >& /dev/null
    RUBY_CONF="/etc/sysconfig/ruby.conf"
else
    action "ruby_init: Configuration file missing" /bin/false
    exit 1
fi

function get_status() {
    run_hackvideo status
#    prefbus_used ?
#    print dumb_con=?
    get_kbds      #and curr. active VTs
}

function get_ext_status() {
    run_input_rc status
    run_hackvideo status
#    prefbus_used ?
#    print dumb_con=?
    get_kbds      #and curr. active VTs
    get_links mouse
    get_links event
}

function run_start_restart() {
    load_modules
    run_hackvideo start
    run_input_rc start
    run_auto_dm
    run_auto_xfree
}

function run_input_rc () {
    for RC in /etc/hotplug/input.rc
    do
        eval "doit=\"\$HOTPLUG_RC_input\""
        if [ "$doit" != yes -a "$1" != status ]; then
            continue
        fi
        $RC $1
        if [ "$1" != status ]; then
            action "bruby: configuring cold-plugged devices." $RC $1

```

```

        fi
    done
}

function run_hackvideo () {
    eval "doit=\"\$HACK_VIDEO_ENABLE\""
    if [ "$doit" != yes ]; then
        if [ "$1" = status ]; then
            echo "XFree hackvideo not configured."
        fi
        return ;
    fi
    if [ -f /proc/bus/pci/hackvideo ]; then
        if [ -f /etc/X11/XF86Config-4 ]; then
            xf_file="/etc/X11/XF86Config-4"
        elif [ -f /etc/X11/XF86Config ]; then
            xf_file="/etc/X11/XF86Config"
        else
            action "bruby: XFree configuration file not
found" /bin/false
            exit 1
        fi
        #fi
        eval "xf_hackvideo=`sed -n '/^#/d;s/^.*"PciOsConfig"[
]//p' $xf_file | sed -e 's/[      ]//g'`"
        if [ "$xf_hackvideo" = "1" ]; then
            case "$1" in
                start)
                    action "bruby: Enabling XFree hackvideo
workaround." /bin/true
                    /bin/echo "1" > /proc/bus/pci/hackvideo
                    ;;
                stop)
                    action "bruby: Disabling XFree
hackvideo workaround." /bin/true
                    /bin/echo "0" > /proc/bus/pci/hackvideo
                    ;;
                status)
                    eval "hack_enabled=`cat
/proc/bus/pci/hackvideo`"
                    if [ $hack_enabled = 1 ]; then
                        echo "XFree hackvideo activated"
                    else
                        echo "XFree hackvideo not
activated,"
                        echo "but enabled in configuration."
                    fi
                ;;
            esac
        fi
    ;;
}

```

```

        *)
            ;;
        esac
    else
        action "bruby: Hackvideo not configured in
XFree," /bin/false
        action "bruby: but enabled in $RUBY_CONF."
/bin/false
    fi
    else
        action "bruby: XFree hackvideo
configured," /bin/false
        action "bruby: but kernel hackvideo
support missing." /bin/false
    fi
}

function load_modules () {
    eval "doit=\"\$LOAD_MODULES\""
    if [ "$doit" != yes -a "$1" != status ]; then
        return ;
    fi
    action "bruby: loading additional modules." /bin/true
    cat $RUBY_CONF | sed -n "/input modules /,/config/p" |
while read module args
do
    case "$module" in
        \#*|") continue ;;
    esac
    initlog -s "Loading module: $module"
    modprobe $module $args >/dev/null 2>&1
done
}

function run_auto_xfree () {
    eval "doit=\"\$AUTO_XFree\""
    if [ "$doit" != yes -a "$1" != status ]; then
        return ;
    fi
    # (pixel) a kind of profile for XF86Config
    # if no XFree=XXX given on kernel command-line, restore
XF86Config.standard
    for i in XF86Config XF86Config-4; do
        if [ -L "/etc/X11/$i" ]; then
            XFree=`sed -n 's/. *XFree=(\w*).*/\1/p'
/proc/cmdline`
            [ -n "$XFree" ] || XFree=standard
            [ -r "/etc/X11/$i.$XFree" ] && ln -sf "$i.$XFree"

```

```

"/etc/X11/$i"
    fi
done
    action "bruby: configuring XFree." /bin/true
}

function run_auto_dm () {
    eval "doit=\"\$AUTO_DM\""
    if [ "$doit" != yes -a "$1" != status ]; then
        return ;
    fi
    #
    #the same like XF86Config but for gdm.conf & Xservers
    #
    for i in xdm/Xservers gdm/gdm.conf; do
        if [ -L "/etc/X11/$i" ]; then
            DumbCon=`sed -n 's/.*dumbcon=\([0-9]*\)*/\1/p'
/proc/cmdline`
            [ -n "$DumbCon" ] || DumbCon=0
            [ -r "/etc/X11/$i.$DumbCon" ] && ln -sf
"/etc/X11/$i.$DumbCon" "/etc/X11/$i"
        fi
    done
    action "bruby: configuring display manager." /bin/true
    echo "Setting up display managers for `expr $DumbCon + 1`
Xservers"
}

function get_kbds() {
    j=0
    for i in /proc/bus/console/*
    do
        echo
        j=`expr $j + 1`
        phys=`cat $i/keyboard`
        if [ "$phys" = "" ];then
            echo " VT-`basename $i` : keyboard not attached"
        else
            echo " VT-`basename $i` using :"
            #
            grep -n2 "$phys" /proc/bus/input/devices | sed -e
'/{/^B: /s;^B:.*;;};{s;^.*: ;;};' -e "/EV=/d"
            grep -B2 -A1 "$phys" /proc/bus/input/devices |
sed 's;^.*: ;;;'
        fi
    done
    echo
    echo "Total of $j VT's avialable."
}

```

```

function get_links() {
    echo
    for i in /dev/input/$1*br
    do
        real=`ls -l $i 2>/dev/null | sed
"{s;^.*\/dev\/input\/;;};{s; ->;}" | cut -d " " -f 2`
        if [ "$real" != "" ]; then
            echo "$i"
            sed -e '/Name/,/Handl/s/.*: //;{/^B:
/s;^B:.*;};' /proc/bus/input/devices | grep -B2 -A1 "$real"
        else
            exit 0
        fi
    done
}

if [ -f /proc/bus/console -o -n tmp=`uname -r | sed -n
's:ruby::p` ]; then
    if [ -x /etc/hotplug/input.rc -a -f /etc/hotplug/input.agent
]; then

        case "$1" in
            start|restart)
                run_start_restart
                ;;
            status)
                get_status
                touch /var/lock/subsys/ruby
                ;;
            ext-status)
                get_ext_status
                touch /var/lock/subsys/ruby
                ;;
            stop)
                #run_hackvideo stop
                run_input_rc stop
                rm -f /var/lock/subsys/ruby
                ;;
            force-reload)
                run_input_rc stop
                run_input_rc start
                touch /var/lock/subsys/ruby
                ;;
            *)
                gprintf "Usage: %s {start|stop|restart|status|
ext-status|force_reload}\n" "$0"
                exit 3
                ;;
        esac
    fi
fi

```

```
        esac

        else if [ -f /etc/hotplug/input.rc -a -f
/etc/hotplug/input.agent ]; then
            action "Input: input.rc and input.agent installed, but
not executable." /bin/false
            action "Input: Please check the file permissions."
/bin/false
        else
            action "Input: Failed to configure cold plugged
devices." /bin/false
            action "Input: input.rc or input.agent missing."
/bin/false
        fi
    fi
fi
```

## Apêndice F - Listagem do arquivo /etc/sysconfig/ruby.conf

```
# /etc/sysconfig/ruby.conf
# This file contains defaults for bruby_init
#
# HACK_VIDEO_ENABLE controls wether the ruby kernel should
# ignore certain XFree commands.
#
# Set to yes in case you are not running the XFree-PrefBusID
# but XFree from your distribution
#
HACK_VIDEO_ENABLE=no

# HOTPLUG_RC_input controls whether the input subsystem is
# started by
# hotplug rc script ("cold plugging")
#
HOTPLUG_RC_input=yes

#
# AUTO_DM controls wether the display manager configuration files
# is adjusted upon the boot argument dumbcon=[number]
#
# AUTO_XFree controls wether the XFree configuration file is
# adjusted upon XFree=[string] argument.
# For Mandrake users this always activated in
/etc/rc.d/rc.sysinit
# and can not be disabled.
#
AUTO_DM=no
AUTO_XFree=no

#
# LOAD_MODULES controls wether the additional modules listed
# below
# should be loaded when a ruby kernel is running.
LOAD_MODULES=yes

#
# input modules configuration
#
# module arguments
# needed for all mice
mousedev
```

```

# for USB input
hid

# for PS2 input
serio
i8042
atkbd
psmouse
keybdrv

#
# keyboard devices configuration
#
# vt_name device_physical_location

VT0      isa0060/serio0/input0
VT1      usb-00:10.0-1/input0
VT2      usb-00:10.0-2/input0
VT3      usb-00:10.1-1/input0

#
# multimedia keys configuration
#
# vt_name device_physical_location
#VT1      usb-00:10.1-1.1/input1
#VT2      usb-*. *-1/input0
#
# mouse devices configuration
#
# sym_link device_physical_location

mouse0    isa0060/serio1/input0
mouse1    usb-00:10.0-1/input1
mouse2    usb-00:10.0-2/input1
mouse3    usb-00:10.1-1/input1

#
# event devices configuration
#
# sym_link device_physical_location
#event0br  isa0060/serio1/input0
#event1br  usb-00:10.1-1.1/input0
#event2br  usb-00:10.1-1.1/input1
#event3br  usb-00:10.1-1.2/input0

#
# end input device configuration
#

```



## Apêndice G - Listagem do arquivo /etc/X11/XFreeConfig1

```
Section "ServerFlags"
    Option      "PciOsConfig" "1"
EndSection

Section "ServerLayout"

#       Indica o servidor gráfico em uso, link para Xfree86-450-
#       prefbusid3
    Identifier   "X1"
    Screen      0  "Screen1" 0 0
    InputDevice "Mouse1"  "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"
#       Option "SingleCard" "true"
EndSection

Section "Files"
    RgbPath    "/usr/X11R6/lib/X11/rgb"
EndSection

Section "Module"
    Load      "dbe"
    Load      "extmod"
    Load      "record"
    Load      "glx"
EndSection

Section "InputDevice"
    Identifier "Keyboard1"
    Driver     "keyboard"
    Option     "XkbRules"  "xfree86"
    Option     "XkbModel"  "abnt2"
    Option     "XkbLayout" "br"
EndSection

Section "InputDevice"
    Identifier "Mouse1"
    Driver     "mouse"
    Option     "Protocol"  "ImPS/2"
    Option     "Device"    "/dev/input/mouse1"
    Option     "ZAxisMapping" "4 5"
    Option     "Emulate3Buttons" "no"
EndSection

Section "Monitor"
    Identifier "Monitor1"
    VendorName "Monitor Vendor"
    ModelName  "500G"
```

```

    HorizSync      30-60
    VertRefresh    43-66
EndSection

Section "Device"
    Identifier      "Videocard1"
    Driver          "nvidia" <<<<<<<<< Driver baixado de
www.nvidia.com
    BusID          "PCI:00:0a:0" <<<<<<<<< Endereço físico da
placa de rede
    BoardName      "RIVA TNT2"
EndSection

Section "Screen"
    Identifier      "Screen1"
    Device          "Videocard1"
    Monitor         "Monitor1"
    DefaultDepth    16

    Subsection "Display"
        Depth        16
        Modes         "1024x768" "800x600" "640x480"
    EndSubsection
EndSection

```

## Apêndice H - Listagem do arquivo /etc/X11/gdm/gdm.conf

```
# GDM Configuration file.  You can use gdmsetup program to
graphically
# edit this, or you can optionally just edit this file by hand.
Note that
# gdmsetup does not tweak every option here, just the ones most
users
# would care about.  Rest is for special setups and distro
specific
# tweaks.  If you edit this file, you should send the HUP or USR1
signal to
# the daemon so that it restarts: (Assuming you have not changed
PidFile)
#   kill -USR1 `cat /var/run/gdm.pid`
# (HUP will make gdm restart immediately while USR1 will make gdm
not kill
# existing sessions and will only restart gdm after all users log
out)
#
# You can also use the gdm-restart and gdm-safe-restart scripts
which just
# do the above for you.
#
# For full reference documentation see the gnome help browser
under
# GNOME|System category.  You can also find the docs in HTML form
# on http://www.jirka.org/gdm.html
#
# NOTE: Some of these are commented out but still show their
default values.
# If you wish to change them you must remove the '#' from the
beginning of
# the line.  The commented out lines are lines where the default
might
# change in the future, so set them one way or another if you
feel
# strongly about it.
#
# Have fun! - George

[daemon]
# Automatic login, if true the first local screen will
automatically logged
# in as user as set with AutomaticLogin key.
AutomaticLoginEnable=false
AutomaticLogin=
```

```

# Timed login, useful for kiosks.  Log in a certain user after a
certain
# amount of time
TimedLoginEnable=false
TimedLogin=
TimedLoginDelay=30

# The gdm configuration program that is run from the login
screen, you should
# probably leave this alone
#Configurator=/usr/sbin/gdmsetup --disable-sound --disable-crash-
dialog
# The chooser program.  Must output the chosen host on stdout,
probably you
# should leave this alone
#Chooser=/usr/bin/gdmchooser
# Greeter for local (non-xdmcp) logins.  Change gdmgreeter to
gdmlogin to
# get the standard greeter.
Greeter=/usr/bin/gdmgreeter
# Greeter for xdmcp logins, usually you want a less graphically
intensive
# greeter here so it's better to leave this with gdmlogin
#RemoteGreeter=/usr/bin/gdmlogin
# Launch the greeter with an additional list of colon seperated
gtk
# modules.  This is useful for enabling additional feature support
# e.g. gnome accessibility framework.  Only "trusted" modules
should
# be allowed to minimise security holes
#AddGtkModules=false
# By default these are the accessibility modules
#GtkModulesList=gail:atk-
bridge:dwellmouselistener:keymouselistener

# Default path to set.  The profile scripts will likely override
this
DefaultPath=/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin
# Default path for root.  The profile scripts will likely
override this
RootPath=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr
/bin:/usr/X11R6/bin

# If you are having trouble with using a single server for a long
time and
# want gdm to kill/restart the server, turn this on
#AlwaysRestartServer=false

# User and group that gdm should run as.  Probably should be gdm

```

```

and gdm and
# you should create these user and group. Anyone found running
this as
# someone too priviliged will get a kick in the ass. This should
have
# access to only the gdm directories and files.
User=gdm
Group=gdm
# To try to kill all clients started at greeter time or in the
Init script.
# doesn't always work, only if those clients have a window of
their own

#KillInitClients=true >>> Valor Original
KillInitClients=true

LogDir=/var/log/gdm
# You should probably never change this value unless you have a
weird setup
PidFile=/var/run/gdm.pid
# Note that a post login script is run before a PreSession
script.
# It is run after the login is successful and before any setup is
# run on behalf of the user
PostLoginScriptDir=/etc/X11/gdm/PostLogin/
PreSessionScriptDir=/etc/X11/gdm/PreSession/
PostSessionScriptDir=/etc/X11/gdm/PostSession/
DisplayInitDir=/etc/X11/gdm/Init
# Distributions: If you have some script that runs an X server
in say
# VGA mode, allowing a login, could you please send it to me?

#FailsafeXServer= >>> Valor Original
FailsafeXServer=

# if X keeps crashing on us we run this script. The default one
does a bunch
# of cool stuff to figure out what to tell the user and such and
can
# run an X configuration program.
XKeepsCrashing=/etc/X11/gdm/XKeepsCrashing
# Reboot, Halt and suspend commands, you can add different
commands
# separated by a semicolon and gdm will use the first one it can
find
#RebootCommand=/usr/bin/reboot;/sbin/reboot;/sbin/shutdown -r
now;/usr/sbin/shutdown -r now
#HaltCommand=/usr/bin/poweroff;/sbin/poweroff;/sbin/shutdown -h

```

```

now;/usr/sbin/shutdown -h now
#SuspendCommand=
# Probably should not touch the below this is the standard setup
ServAuthDir=/var/gdm
# This is our standard startup script.  A bit different from a
normal
# X session, but it shares a lot of stuff with that.  See the
provided
# default for more information.
BaseXsession=/etc/X11/xdm/Xsession
# This is a directory where .desktop files describing the
sessions live
# It is really a PATH style variable since 2.4.4.2 to allow
actual
# interoperability with KDM
#SessionDesktopDir=/etc/X11/sessions/;/etc/X11/dm/Sessions/;/usr/
share/xsessions/
# This is the default .desktop session.  One of the ones in
SessionDesktopDir
DefaultSession=default.desktop
# Better leave this blank and HOME will be used.  You can use
syntax ~/ below
# to indicate home directory of the user.  You can also set this
to something
# like /tmp if you don't want the authorizations to be in home
directories.
# This is useful if you have NFS mounted home directories.  Note
that if this
# is the home directory the UserAuthFBDir will still be used in
case the home
# directory is NFS, see security/NeverPlaceCookiesOnNFS to
override this behaviour.
UserAuthDir=
# Fallback if home directory not writable
UserAuthFBDir=/tmp
UserAuthFile=.Xauthority
# The X server to use if we can't figure out what else to run.

#StandardXServer=/usr/X11R6/bin/X    >>> VALOR ORIGINAL
StandardXServer=/usr/X11R6/bin/Xl

# The maximum number of flexible X servers to run.

#FlexibleXServers=5    >>> VALOR ORIGINAL
FlexibleXServers=5

# the X nest command
Xnest=/usr/X11R6/bin/Xnest -audit 0 -name Xnest
# Automatic VT allocation.  Right now only works on Linux.  This

```

```

way
# we force X to use specific vts.  turn VTAllocation to false if
this
# is causing problems.

#FirstVT=7 >>> VALOR ORIGINAL
#VTAllocation=true >>> VALOR ORIGINAL

FirstVT=7
VTAllocation=true

# Should double login be treated with a warning (and possibility
to change
# vts on linux systems for console logins)
#DoubleLoginWarning=true

[security]
# If any distributions ship with this one off, they should be
shot
# this is only local, so it's only for say kiosk use, when you
# want to minimize possibility of breakin
AllowRoot=true
# If you want to be paranoid, turn this one off
AllowRemoteRoot=true
# This will allow remote timed login
AllowRemoteAutoLogin=false
# 0 is the most anal, 1 allows group write permissions, 2 allows
all write
# permissions
RelaxPermissions=0
# Number of seconds to wait after a bad login
RetryDelay=1
# Maximum size of a file we wish to read.  This makes it hard for
a user to DoS
# us by using a large file.
UserMaxFile=65536
# If true this will basically append -nolisten tcp to every X
command line,
# a good default to have (why is this a "negative" setting?
because if
# it is false, you could still not allow it by setting command
line of
# any particular server).  It's probably better to ship with this
on
# since most users will not need this and it's more of a security
risk
# then anything else.
# Note: Anytime we find a -query or -indirect on the command line
we do

```

```

# not add a "-nolisten tcp", as then the query just wouldn't
work, so
# this setting only affects truly local sessions.
#DisallowTCP=true
# By default never place cookies if we "detect" NFS. We detect
NFS
# by detecting "root-squashing". It seems bad practice to place
# cookies on things that go over the network by default and thus
we
# don't do it by default. Sometimes you can however use safe
remote
# filesystems where this is OK and you may want to have the
cookie in your
# home directory.
#NeverPlaceCookiesOnNFS=true

# XDMCP is the protocol that allows remote login. If you want to
log into
# gdm remotely (I'd never turn this on on open network, use ssh
for such
# remote usage that). You can then run X with -query <thishost>
to log in,
# or -indirect <thishost> to run a chooser. Look for the
'Terminal' server
# type at the bottom of this config file.
[xdmcp]
# Distributions: Ship with this off. It is never a safe thing to
leave
# out on the net. Setting up /etc/hosts.allow and
/etc/hosts.deny to only
# allow local access is another alternative but not the safest.
# Firewalling port 177 is the safest if you wish to have xdmcp
on.
# Read the manual for more notes on the security of XDMCP.
Enable=false
# Honour indirect queries, we run a chooser for these, and then
redirect
# the user to the chosen host. Otherwise we just log the user in
locally.
HonorIndirect=true
# Maximum pending requests
#MaxPending=4
#MaxPendingIndirect=4
# Maximum open XDMCP sessions at any point in time
#MaxSessions=16
# Maximum wait times
#MaxWait=15
#MaxWaitIndirect=15
# How many times can a person log in from a single host. Usually

```



```

better to
# keep low to fend off DoS attacks by running many logins from a
single
# host. This is now set at 2 since if the server crashes then
gdm doesn't
# know for some time and wouldn't allow another session.
#DisplaysPerHost=2
# The number of seconds after which a non-responsive session is
logged off.
# Better keep this low.
#PingIntervalSeconds=15
# The port. 177 is the standard port so better keep it that way
#Port=177
# Willing script, none is shipped and by default we'll send
# hostname system id. But if you supply something here, the
# output of this script will be sent as status of this host so
that
# the chooser can display it. You could for example send load,
# or mail details for some user, or some such.
Willing=/etc/X11/gdm/Xwilling

[gui]
# The 'theme'. By default we're using the default gtk theme
# Of course assuming that gtk got installed in the same prefix,
# if not change this.
GtkRC=/usr/share/themes/Bluecurve/gtk-2.0/gtkrc
# Maximum size of an icon, larger icons are scaled down
#MaxIconWidth=128
#MaxIconHeight=128

[greeter]
# Greeter has a nice title bar that the user can move
TitleBar=false
# Configuration is available from the system menu of the greeter
ConfigAvailable=false
# Face browser is enabled. This only works currently for the
# standard greeter as it is not yet enabled in the graphical
greeter.
Browser=false
# The default picture in the browser
#DefaultFace=/usr/share/pixmaps/nobody.png
# These are things excluded from the face browser, not from
logging in
#Exclude=bin,daemon,adm,lp,sync,shutdown,halt,mail,news,uucp,oper
ator,nobody,gdm,postgres,pvm,rpm,nfsnobody,pcap
# As an alternative to the above this is the minimum uid to show
MinimalUID=500
# If user or user.png exists in this dir it will be used as his
picture

```

```
#GlobalFaceDir=/usr/share/faces/
# File which contains the locale we show to the user. Likely you
# want to use
# the one shipped with gdm and edit it. It is not a standard
# locale.alias file,
# although gdm will be able to read a standard locale.alias file
# as well.
#LocaleFile=/etc/X11/gdm/locale.alias
# Logo shown in the standard greeter
#Logo=/usr/share/pixmaps/gdm-foot-logo.png
Logo=
## nice RH logo for the above line:
/usr/share/pixmaps/redhat/shadowman-200.png
# The standard greeter should shake if a user entered the wrong
# username or
# password. Kind of cool looking
#Quiver=true
# The Actions menu (formerly system menu) is shown in the
# greeter, this is the
# menu that contains reboot, shutdown, suspend, config and
# chooser. None of
# these is available if this is off. They can be turned off
# individually
# however
#SystemMenu=true
# Should the chooser button be shown. If this is shown, GDM can
# drop into
# chooser mode which will run the xdmcp chooser locally and allow
# the user
# to connect to some remote host. Local XDMCP does not need to
# be enabled
# however
#ChooserButton=true
# Note to distributors, if you wish to have a different Welcome
# string
# and wish to have this translated you can have entries such as
# Welcome[cs]=Vítejte na %n
# Just make sure the string is in utf-8
# Welcome is for all console logins and RemoteWelcome is for
# remote logins
# (through XDMCP).
# The default entries that are shipped are translated inside
# genius and
# are as follows:
#Welcome=Welcome
#RemoteWelcome=Welcome to %n
# Don't allow user to move the standard greeter window. Only
# makes sense
# if TitleBar is on
```

```
#LockPosition=false
# Set a position rather than just centering the window.  If you
enter
# negative values for the position it is taken as an offset from
the
# right or bottom edge.
#SetPosition=false
#PositionX=0
#PositionY=0
# Xinerama screen we use to display the greeter on.  Not for true
# multihead, currently only works for Xinerama.
#XineramaScreen=0
# Background settings for the standard greeter:
# Type can be 0=None, 1=Image, 2=Color
#BackgroundType=2
#BackgroundImage=
#BackgroundScaleToFit=true
BackgroundColor=#20305a
# XDMCP session should only get a color, this is the sanest
setting since
# you don't want to take up too much bandwidth
#BackgroundRemoteOnlyColor=true
# Program to run to draw the background in the standard greeter.
Perhaps
# something like an xscreensaver hack or some such.
#BackgroundProgram=
# if this is true then the background program is run always,
otherwise
# it is only run when the BackgroundType is 0 (None)
#RunBackgroundProgramAlways=false
# Show the Failsafe sessions.  These are much MUCH nicer (focus
for xterm for
# example) and more failsafe than those supplied by scripts so
distros should
# use this rather than just running an xterm from a script.
ShowGnomeFailsafeSession=false
#ShowXtermFailsafeSession=true
# Normally there is a session type called 'Last' that is shown
which refers to
# the last session the user used.  If off, we will be in
'switchdesk' mode where
# the session saving stuff is disabled in GDM
ShowLastSession=false
# Always use 24 hour clock no matter what the locale.
#Use24Clock=false
# Use circles in the password field.  Looks kind of cool
actually,
# but only works with certain fonts.
#UseCirclesInEntry=false
```

```

# These two keys are for the new greeter.  Circles is the
standard
# shipped theme
GraphicalTheme=Bluecurve
GraphicalThemeDir=/usr/share/gdm/themes/

# The chooser is what's displayed when a user wants an indirect
XDMCP
# session, or selects Run XDMCP chooser from the system menu
[chooser]
# Default image for hosts
#DefaultHostImg=/usr/share/pixmaps/nohost.png
# Directory with host images, they are named by the hosts: host
or host.png
HostImageDir=/usr/share/hosts/
# Time we scan for hosts (well only the time we tell the user we
are
# scanning actually, we continue to listen even after this has
# expired)
#ScanTime=4
# A comma separated lists of hosts to automatically add (if they
answer to
# a query of course).  You can use this to reach hosts that
broadcast cannot
# reach.
Hosts=
# Broadcast a query to get all hosts on the current network that
answer
#Broadcast=true
# Allow adding random hosts to the list by typing in their names
#AllowAdd=true

[debug]
# This will enable debugging into the syslog, usually not
neccessary
# and it creates a LOT of spew of random stuff to the syslog.
However it
# can be useful in determining when something is going very
wrong.
Enable=false

[servers]
# These are the standard servers.  You can add as many you want
here
# and they will always be started.  Each line must start with a
unique
# number and that will be the display number of that server.
Usually just
# the 0 server is used.

```

```

0=Standard
### INSERIDOS
1=2nd
2=3nd
3=4nd

# Note the VTAllocation and FirstVT keys on linux.  Don't add any
vt<number>
# arguments if VTAllocation is on, and set FirstVT to be the
first vt
# available that your gettys don't grab (gettys are usually dumb
and grab
# even a vt that has already been taken).  Using 7 will work
pretty much for
# all linux distributions.  VTAllocation is not currently
implemented on
# anything but linux since I don't own any non-linux systems.
Feel free to
# send patches.  X servers will just not get any extra arguments
then.
#
# If you want to run an X terminal you could add an X server such
as this
#0=Terminal -query serverhostname
# or for a chooser (optionally serverhostname could be localhost)
#0=Terminal -indirect serverhostname
#
# If you wish to run the XDMCP chooser on the local display use
the following
# line
#0=Chooser

## Note:
# is your X server not listening to TCP requests?  Perhaps you
should look
# at the security/DisallowTCP setting!

# Definition of the standard X server.
[server-Standard]
name=Standard server
### command=/usr/X11R6/bin/X :0 -xf86config /etc/X11/XF86Config
-deferglyphs 16 -ac vt7 >>> ORIGINAL
command=/usr/X11R6/bin/X0 :0 -xf86config /etc/X11/XF86Config-0
-deferglyphs 16 -ac vt7
flexible=true

## INSERIDAS REFERÊNCIA PARA OS NOVOS TERMINAIS
[server-2nd]

```

```

name=2nd server
command=/usr/X11R6/bin/X1 :1 -xf86config /etc/X11/XF86Config-1
-deferglyphs 16 vt17
flexible=true

[server-3nd]
name=3nd server
command=/usr/X11R6/bin/X2 :2 -xf86config /etc/X11/XF86Config-2
-deferglyphs 16 vt18
flexible=true

[server-4nd]
name=4nd server
command=/usr/X11R6/bin/X3 :3 -xf86config /etc/X11/XF86Config-3
-deferglyphs 16 vt19
flexible=true

# To use this server type you should add -query host or -indirect
host
# to the command line
[server-Terminal]
name=Terminal server
# Add -terminate to make things behave more nicely
command=/usr/X11R6/bin/X -audit 0 -terminate
# Make this not appear in the flexible servers (we need extra
params
# anyway, and terminate would be bad for xdmcp choosing). You
can
# make a terminal server flexible, but not with an indirect
query.
# If you need flexible indirect query server, then you must get
rid
# of the -terminate and the only way to kill the flexible server
will
# then be by Ctrl-Alt-Backspace
flexible=false
# Not local, we do not handle the logins for this X server
handled=false

# To use this server type you should add -query host or -indirect
host
# to the command line
[server-Chooser]
name=Chooser server
command=/usr/X11R6/bin/X -audit 0
# Make this not appear in the flexible servers for now, but if
you
# wish to allow a chooser server then make this true. This is
the

```

```
# only way to make a flexible chooser server that behaves nicely.
flexible=false
# Run the chooser instead of the greeter.  When the user chooses
a
# machine they will get this same server but run with
# "-terminate -query hostname"
chooser=true
```

## Apêndice I - Listagem completa de processos em execução

```
USER PID STAT START COMMAND
root 1 S 13:19 init
root 2 SW 13:19 [keventd]
root 3 SWN 13:19 [ksoftirqd_CPU0]
root 4 SW 13:19 [kswapd]
root 5 SW 13:19 [bdflush]
root 6 SW 13:19 [kupdated]
root 8 SW 13:19 [khubd]
root 14 SW 13:19 [kseriod]
root 17 SW 13:19 [kjournald]
root 22 S 13:19 /sbin/devfsd
root 723 SW 13:19 [kjournald]
root 725 SW 13:19 [kjournald]
root 734 SW 13:19 [kjournald]
root 2857 S 13:19 syslogd
root 2861 S 13:19 klogd
rpc 2882 S 13:19 portmap
rpcuser 2902 S 13:19 rpc.statd
root 2989 S 13:19 /usr/sbin/smardt
root 3171 S 13:19 /usr/sbin/sshd
root 3187 S 13:19 xinetd
root 3200 S 13:19 /bin/sh
mysql 3224 S 13:19 /usr/libexec/mysqld
mysql 3225 S 13:19 /usr/libexec/mysqld
mysql 3226 S 13:19 /usr/libexec/mysqld
mysql 3227 S 13:19 /usr/libexec/mysqld
root 3257 S 13:19 sendmail:
smmsp 3266 S 13:19 sendmail:
root 3277 S 13:19 gpm
root 3291 S 13:19 /usr/sbin/httpd
root 3301 S 13:19 crond
xfs 3325 S 13:19 xfs
daemon 3344 S 13:19 /usr/sbin/atd
dbus 3354 S 13:19 dbus-daemon-1
root 3377 S 13:19 /sbin/mingetty
root 3378 S 13:19 /usr/bin/gdm-binary
root 3407 S 13:19 xdm
root 3422 S 13:19 /usr/bin/gdm-binary
root 3423 SL 13:19 /usr/X11R6/bin/X0
root 3429 S 13:20 /usr/bin/gdm-binary
root 3436 SL 13:20 /usr/X11R6/bin/X1
root 3440 S 13:20 /usr/bin/gdm-binary
root 3448 SL 13:20 /usr/X11R6/bin/X2
root 3452 S 13:20 /usr/bin/gdm-binary
root 3460 SL 13:20 /usr/X11R6/bin/X3
gdm 3470 S 13:20 /usr/bin/gdmgreeter
```



```
root 3473 S 13:20 -19
root 3479 S 13:21 -19
547 3824 S 13:38 /bin/sh
547 3878 S 13:38 /usr/bin/ssh-agent
547 3909 S 13:38 kdeinit:
547 3912 S 13:38 kdeinit:
547 3915 S 13:38 kdeinit:
547 3918 S 13:38 kdeinit:
547 3925 S 13:38 /usr/bin/artsd
547 3951 S 13:38 kdeinit:
547 3952 S 13:38 kwrapper
547 3954 S 13:38 kdeinit:
547 3955 S 13:38 kdeinit:
547 3957 S 13:38 kdeinit:
547 3959 S 13:38 kdeinit:
547 3993 S 13:38 kdeinit:
547 3994 S 13:38 /usr/bin/pam-panel-icon
root 3995 S 13:38 /sbin/pam_timestamp_check
547 4051 S 13:39 /bin/sh
547 4070 S 13:39 /bin/sh
547 4075 R 13:39 /usr/lib/firefox-1.0.4/firefox-bin
547 4077 S 13:39 /usr/lib/firefox-1.0.4/firefox-bin
547 4078 S 13:39 /usr/lib/firefox-1.0.4/firefox-bin
547 4080 S 13:39 /usr/libexec/gconfd-2
547 4082 S 13:39 /usr/lib/firefox-1.0.4/firefox-bin
549 4137 S 13:40 /bin/sh
549 4191 S 13:40 /usr/bin/ssh-agent
549 4222 S 13:40 kdeinit:
549 4225 S 13:40 kdeinit:
549 4228 S 13:40 kdeinit:
549 4231 S 13:40 kdeinit:
549 4241 S 13:40 /usr/bin/artsd
549 4274 S 13:40 kdeinit:
549 4275 S 13:40 kwrapper
549 4277 S 13:40 kdeinit:
549 4278 S 13:40 kdeinit:
549 4280 S 13:40 kdeinit:
549 4282 S 13:40 kdeinit:
549 4283 S 13:41 kdeinit:
549 4287 S 13:41 kdeinit:
549 4288 S 13:41 /usr/bin/pam-panel-icon
549 4289 S 13:41 eggcups
root 4290 S 13:41 /sbin/pam_timestamp_check
549 4293 S 13:41 /usr/libexec/gconfd-2
549 4294 S 13:41 /usr/libexec/evolution/1.4/evolution-alarm-
notify
549 4297 SN 13:41 /usr/bin/python
549 4299 S 13:41 /usr/libexec/bonobo-activation-server
549 4325 S 13:41 bluefish
```

```
543 4327 S 13:41 /bin/sh
543 4385 S 13:41 /usr/bin/ssh-agent
543 4416 S 13:41 kdeinit:
543 4419 S 13:41 kdeinit:
543 4422 S 13:41 kdeinit:
543 4426 S 13:41 kdeinit:
543 4435 S 13:41 /usr/bin/artsd
543 4461 S 13:41 kdeinit:
543 4462 S 13:41 kwrapper
543 4464 S 13:41 kdeinit:
543 4465 S 13:41 kdeinit:
543 4467 S 13:41 kdeinit:
543 4469 S 13:41 kdeinit:
543 4470 S 13:41 kdeinit:
543 4475 S 13:41 kdeinit:
543 4476 S 13:41 /usr/bin/pam-panel-icon
root 4477 S 13:41 /sbin/pam_timestamp_check
543 4478 S 13:41 eggcups
543 4479 SN 13:41 /usr/bin/python
543 4481 S 13:41 /usr/libexec/gconfd-2
549 4668 S 13:42 /bin/sh
549 4687 S 13:42 /bin/sh
549 4692 S 13:42 /usr/lib/firefox-1.0.4/firefox-bin
549 4694 S 13:42 /usr/lib/firefox-1.0.4/firefox-bin
549 4695 S 13:42 /usr/lib/firefox-1.0.4/firefox-bin
549 4697 S 13:42 /usr/lib/firefox-1.0.4/firefox-bin
547 4712 S 13:44 bluefish
543 4771 S 13:52 bluefish
dionerf 4826 S 13:53 /bin/sh
dionerf 4884 S 13:53 /usr/bin/ssh-agent
dionerf 4915 S 13:53 kdeinit:
dionerf 4918 S 13:53 kdeinit:
dionerf 4921 S 13:53 kdeinit:
dionerf 4924 S 13:53 kdeinit:
dionerf 4933 S 13:53 /usr/bin/artsd
dionerf 4959 S 13:53 kdeinit:
dionerf 4960 S 13:54 kwrapper
dionerf 4962 S 13:54 kdeinit:
dionerf 4963 S 13:54 kdeinit:
dionerf 4965 S 13:54 kdeinit:
dionerf 4967 S 13:54 kdeinit:
dionerf 4973 S 13:54 kdeinit:
dionerf 4974 S 13:54 /usr/bin/pam-panel-icon
root 4975 S 13:54 /sbin/pam_timestamp_check
dionerf 4976 S 13:54 eggcups
dionerf 4977 SN 13:54 /usr/bin/python
dionerf 4979 S 13:54 /usr/libexec/gconfd-2
dionerf 4982 S 13:54 /usr/libexec/evolution/1.4/evolution-alarm-notify
```

```
dionerf 4984 S 13:54 /usr/libexec/bonobo-activation-server
dionerf 4985 S 13:54 bluefish
dionerf 4988 S 13:55 /bin/sh
dionerf 5007 S 13:55 /bin/sh
dionerf 5012 S 13:55 /usr/lib/firefox-1.0.4/firefox-bin
dionerf 5014 S 13:55 /usr/lib/firefox-1.0.4/firefox-bin
dionerf 5015 S 13:55 /usr/lib/firefox-1.0.4/firefox-bin
dionerf 5017 S 13:55 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5603 S 14:09 /bin/sh
glauberb 5661 S 14:09 /usr/bin/ssh-agent
glauberb 5692 S 14:09 kdeinit:
glauberb 5695 S 14:09 kdeinit:
glauberb 5700 S 14:09 kdeinit:
glauberb 5703 S 14:09 kdeinit:
glauberb 5712 S 14:10 /usr/bin/artsd
glauberb 5733 S 14:10 kdeinit:
glauberb 5741 S 14:10 kwrapper
glauberb 5743 S 14:10 kdeinit:
glauberb 5744 S 14:10 kdeinit:
glauberb 5746 S 14:10 bluefish
glauberb 5748 S 14:10 kdeinit:
glauberb 5750 S 14:10 kdeinit:
glauberb 5752 S 14:10 /usr/bin/autorun
glauberb 5757 S 14:10 kdeinit:
glauberb 5758 S 14:10 /usr/bin/pam-panel-icon
glauberb 5759 S 14:10 eggcups
root 5760 S 14:10 /sbin/pam_timestamp_check
glauberb 5761 SN 14:10 /usr/bin/python
glauberb 5763 S 14:10 /usr/libexec/gconfd-2
glauberb 5766 S 14:10 /usr/libexec/evolution/1.4/evolution-alarm-
notify
glauberb 5768 S 14:10 /usr/libexec/bonobo-activation-server
glauberb 5771 S 14:10 /bin/sh
glauberb 5790 S 14:10 /bin/sh
glauberb 5795 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5797 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5798 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5800 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
dionerf 5880 S 14:13 kdeinit:
glauberb 5757 S 14:10 kdeinit:
glauberb 5758 S 14:10 /usr/bin/pam-panel-icon
glauberb 5759 S 14:10 eggcups
root 5760 S 14:10 /sbin/pam_timestamp_check
glauberb 5761 SN 14:10 /usr/bin/python
glauberb 5763 S 14:10 /usr/libexec/gconfd-2
glauberb 5766 S 14:10 /usr/libexec/evolution/1.4/evolution-alarm-
notify
glauberb 5768 S 14:10 /usr/libexec/bonobo-activation-server
glauberb 5771 S 14:10 /bin/sh
```

```
glauberb 5790 S 14:10 /bin/sh
glauberb 5795 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5797 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5798 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
glauberb 5800 S 14:10 /usr/lib/firefox-1.0.4/firefox-bin
dionerf 5880 S 14:13 kdeinit:
glauberb 5892 S 14:13 kdeinit:
547 5899 S 14:14 kdeinit:
dionerf 5919 S 14:14 kdeinit:
glauberb 5921 S 14:14 kdeinit:
547 5923 S 14:14 kdeinit:
543 5935 S 14:15 kdeinit:
549 5937 S 14:15 kdeinit:
dionerf 5968 S 14:16 kdeinit:
root 6463 SN 14:24 cupsd
apache 6664 S 14:25 /usr/sbin/httpd
apache 6665 S 14:25 /usr/sbin/httpd
apache 6666 S 14:25 /usr/sbin/httpd
apache 6667 S 14:25 /usr/sbin/httpd
apache 6668 S 14:25 /usr/sbin/httpd
apache 6669 S 14:25 /usr/sbin/httpd
apache 6670 S 14:25 /usr/sbin/httpd
apache 6671 S 14:25 /usr/sbin/httpd
543 13360 S 15:39 /bin/sh
543 13379 S 15:39 /bin/sh
543 13384 S 15:39 /usr/lib/firefox-1.0.4/firefox-bin
543 13386 S 15:39 /usr/lib/firefox-1.0.4/firefox-bin
543 13387 S 15:39 /usr/lib/firefox-1.0.4/firefox-bin
543 13389 S 15:39 /usr/lib/firefox-1.0.4/firefox-bin
root 13601 S 16:31 sshd:
celso 13605 S 16:31 sshd:
celso 13606 S 16:31 -bash
547 13735 S 16:42 /usr/lib/firefox-1.0.4/firefox-bin
549 13742 S 16:43 /usr/lib/firefox-1.0.4/firefox-bin
celso 13743 R 16:43 ps
celso 13744 S 16:43 awk
```