GabiX Linux

Manual do desenvolvedor

Gabriel Marques
snortt@gmail.com

11 de junho de 2013

Resumo

Este documento mostra como utilizar, ajustar e "*hackear*" o GabiX Linux, através de seus arquivos de configurações e *scripts* de inicialização. A configuração e substituição do núcleo do sistema também é coberta em detalhes.

Este documento assume que o leitor possui conhecimentos de nível intermediário/avançado sobre o sistema operacional Linux, além de noções sobre os seguintes assuntos: inicialização do Linux, compilação de programas, aplicação de *patches*, transposição e união de sistemas de arquivos, *shell scripts*, *Makefiles*, gerenciamento de pacotes *.deb* e inicialização do Debian.

Toda vez que este documento utilizar a palavra "GabiX", leia-se *Kernel + initrd + modulos + SquashFS*. Para a palavra "Distribuição", leia-se *Debian GNU/Linux*, uma vez que o GabiX não é uma distribuição Linux, principalmente por se tratar de um sistema do tipo *live*.

Sumário

Lista de Figuras

1 Sobre o Gabix

GabiX é uma ferramenta que permite modularizar e personalizar o processo de inicialização de uma distribuição Linux. Com ele é possível carregar um núcleo Linux, localizar e montar um sistema de arquivos em memória e inicializar algum sistema Linux contido nele. Isso permite ao usuário criar seu próprio sistema Linux em CD/DVD-ROM (*live boot*).

O sistema permite que o usuário carregue suas próprias configurações, previamente armazenadas em mídias removíveis, ou ainda, que ele carregue uma distribuição Linux previamente instalada no disco local da máquina, porém utilizando o *kernel* e os módulos que acompanham seu sistema *live boot*.

O GabiX foi desenvolvido utilizando o sistema operacional Debian GNU/Linux. Se ajustadas de acordo com o descrito neste documento, outras distribuições Linux também podem exercer a mesma função do Debian neste sistema. O Debian GNU/Linux foi escolhido devido à sua estabilidade, praticidade. abrangência e facilidade de manuseio. Embora o GabiX seja apenas um *carregador* para o núcleo (e um sistema de montagem de imagens *live*), a distribuição Linux escolhida também precisou de alguns ajustes para funcionar dentro do ambiente criado. Tal processo também será detalhado ao longo deste documento.

2 Estrutura do GabiX

O GabiX é composto por um núcleo Linux, módulos, firmware, includes de núcleo, *scripts* de inicialização e uma distribuição Linux, modificada e armazenada em um sistema de arquivos do tipo *SquashFS*. A maneira como as partes são sequenciadas e integradas permite a modularização do sistema como um todo, composto por pequenos "módulos" que trabalham em conjunto, utilizando movimentação de pontos de montagens e sobreposição transparente de sistemas de arquivos via *UnionFS*. Essa abordagem permite que o GabiX seja desmontado em partes que podem ser modificadas e/ou substituídas e, posteriormente, agrupadas novamente e gravadas em mídia inicializável.

O processo de integração entre as partes será apresentado detalhadamente ao longo deste documento. A seguir, uma descrição de cada uma dos partes ("módulos") que compõem o GabiX.

- Diretório */boot*: Contém os arquivos de inicialização do sistema: o gestor de boot, o núcleo Linux e o *initrd*.

- Núcleo (vmlinuz): Composto pelo kernel e seus módulos essenciais ao carregamento do sistema, o núcleo é o responsável por inicializar o sistema, detectar o hardware e carregar a próxima etapa, a imagem *initrd*. O arquivo que representa o núcleo tem nome vmlinuz-*versão_do_núcleo*. O nome vmlinuz indica uma imagem de núcleo com suporte à memória virtual. Ela é compactada com gzip[1], bzip2[2], ou ainda, lzma[3].

- *initrd*: Contém um sistema mínimo, baseado em *busybox*, capaz de localizar e montar dispositivos de armazenamento e carregar a próxima etapa, o sistema operacional contido na imagem *SquashFS*. Aqui também são encontrados alguns módulos básicos, necessários para que o núcleo consiga trabalhar, incluindo os módulos *loop*, *SquashFS* e *UnionFS*, além do arquivo *init* e seu arquivo de funções. O arquivo que representa a imagem *initrd* tem nome initrd.img-*versão_do_núcleo*. No GabiX, os termos *initrd* e *miniroot* são intercambiáveis.

- *init*: Contido na imagem *initrd*, é o responsável pelas operações pré-carregamento do sistema em *SquashFS* e por passar o controle do sistema para a distribuição Linux contida na mídia.

- Diretório */base*: Contém as imagens *SquashFS* que armazenam os módulos restantes do núcleo, *firmware* e a distribuição Linux a ser carregada. Cada arquivo neste diretóri é nomeado de acordo com seu propósito, dispensando maiores informações até o presente momento.

[1] www.gzip.org
[2] www.bzip.org
[3] www.7-zip.org, tukaani.org/xz/format.html

3 O processo de inicialização

3.0.1 Sobre a inicialização do Linux

Explicar e detalhar o processo de inicialização do Linux está além do escopo deste documento. Uma leitura complementar sobre "Inicialização do Linux" pode ser facilmente encontrada na Internet e, portanto, este manual apenas descreverá o processo de maneira simplificada.

Antes de começar a desenvolver sistemas *live* usando o GabiX, é necessário um entendimento básico de como funciona o processo de carregamento de um sistema Linux instalado em um computador. De maneira geral, tal processo segue as seguintes etapas:

- O gestor de boot localiza o *kernel* e o executa.
- O *kernel* é extraído para a memória e faz diversas verificações iniciais e checagem de parâmetros.
- Após sua descompressão e ajustes iniciais, ele procura o *initrd* e o extrai para que seu */init* seja executado. Nesta etapa serão tratados módulos adicionais e os acertos no ambiente.
- O initrd prepara o ambiente inicial[4] para, posteriormente, passar o controle[5] para o */sbin/init* da distribuição Linux.
- A distribuição continua seu carregamento, iniciando os serviços solicitados, até imprimir a tela de entrada, seja ela em modo texto ou gráfico.

Mantenha essas etapas em mente enquanto estiver desenvolvendo com o GabiX.

3.0.2 Sobre a inicialização do GabiX

Uma vez entendido o *boot* de um sistema Linux no computador, fica muito fácil entender as pequenas diferenças que caracterizam o mesmo processo, porém usado para carregar um sistema do tipo *live boot*. As seguintes etapas compõem o processo de inicialização do GabiX, desconsiderando o processamento de parâmetros especiais[6] na linha de inicialização do sistema:

1. O Gestor de BOOT carrega o núcleo.
2. O núcleo do Linux descompacta em RAM, localiza e carrega o initrd.
3. Uma vez no initrd, o init faz ajustes iniciais, montando sistemas de arquivos virtuais e populando o diretório */dev*. Depois, ele localiza e monta a unidade de CD/DVD-ROM em */cdrom*.
4. Uma área de memória é montada no diretório /dynamic, usando tmpfs, para operações de escrita no sistema de arquivos. Devido à sobreposição transparente, a escrita é realizada em memória como se fosse no sistema de arquivos real, porém é volátil.
5. Uma vez montada a mídia, é possível carregar o sistema em etapas.
   O init monta o arquivo */cdrom/base/sistema* em */static*.
6. O arquivo */cdrom/base/modulos* é montado em */static/lib/modules*. Este contém os módulos adicionais do núcleo.
7. O arquivo */cdrom/base/firmware* é montado em */static/lib/firmware*. Este contém os "firmware" do núcleo.
8. O arquivo */cdrom/base/headers*, caso exista, é montado em */static/usr/src*. Este contém os "headers" do núcleo (para desenvolvimento).
9. O arquivo */cdrom/base/includes*, caso exista, é montado em */static/usr/include*. Este contém os "includes" do núcleo (para desenvolvimento).

[4] *busy box*
[5] Troca de raiz - *Pivot root*
[6] Parâmetros do kernel serão discutidos posteriormente

10. O diretório */cdrom/boot* é ligado ao diretório */static/boot*, o que o torna visível após o carregamento total do sistema.

11. O ponto de montagem */cdrom* é movido para */static/cdrom*, o que o torna visível após o carregamento total do sistema.

12. A área dinâmica */dynamic* é unida à área estática */static* no ponto de fusão */union*, usando *UnionFS*. Isso permite operações voláteis de escrita no sistema de arquivos, o que faz com que todo o Linux possa ser modificado livremente durante seu funcionamento.

13. O init realiza a troca do sistema de arquivos raiz criado pelo *miniroot* para */union* e executa o arquivo */sbin/init* real da distribuição Linux configurada no *live boot*.

Esta abordagem permite que um núcleo 100% funcional, seus módulos, *firmware*, cabeçalhos (*headers*) e includes sejam carregados e fiquem visíveis para o usuário, mesmo após o carregamento do sistema, que realiza uma troca do ponto de montagem da raiz do sistema. Isso também permite inicializar um sistema Linux instalado em um computador, mesmo quando seu gestor de BOOT e/ou núcleo estejam danificados, desde que o sistema a ser carregado seja de uma arquitetura compatível com o núcleo contido na mídia utilizada. Em suma, esta modularização permite carregar o sistema da mídia com o núcleo da mídia, ou ainda o núcleo da mídia removível com um sistema em um disco. Em casos extremos, é possível também carregar o núcleo da mídia removível e nenhum sistema (seja na mídia removível ou no disco), usando apenas os comandos disponíveis no *miniroot*.

3.0.3 Diagrama de inicialização do GabiX

A imagem a seguir, ilustra o processo utilizado pelo GabiX para iniciar a distribuição Linux.

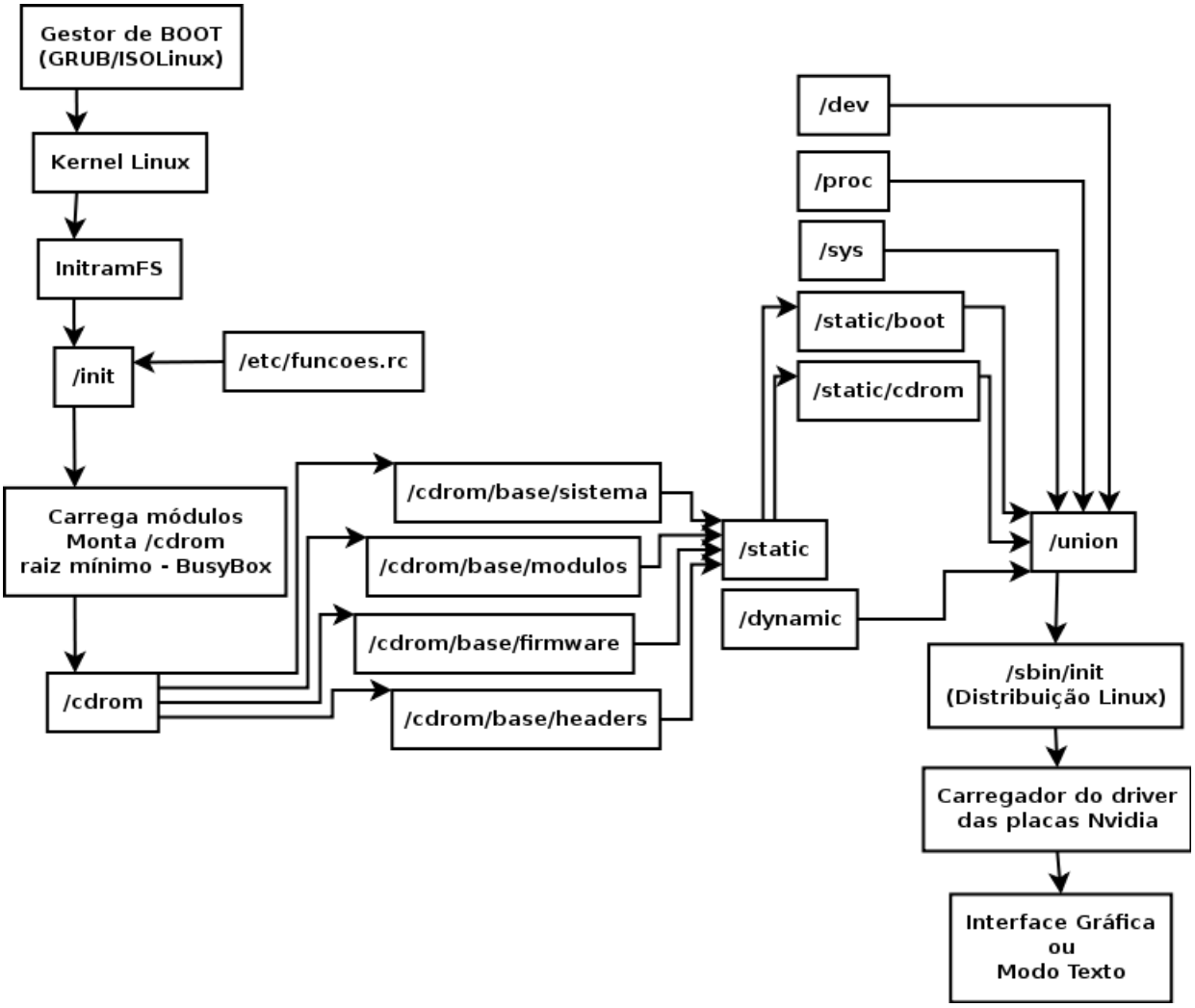

Figura 1: Estrutura do GabiX

3.1 Termos específicos utilizados neste documento

- gabix: é o diretório que dá origem à raiz da imagem ISO gerada pelo GabiX (pode ser um *link* simbólico).
- miniroot: é o diretório que dá origem ao initrd usado pelo GabiX (pode ser um *link* simbólico).
- root-NFS: é o diretório que contém o sistema alvo, a"jaula" com a distribuição Linux usada pelo GabiX. Mais detalhes sobre root-NFS são apresentados em "Sobre a distribuição Linux utilizada" (pode ser um *link* simbólico).
- Núcleo (*kernel*): é o *kernel* Linux utilizado.
- initrd: é o arquivo que contém o sistema mínimo inicial.
- Sistema hospedeiro: é o sistema utilizado para construir o GabiX
- Sistema alvo: é a distribuição Linux- Debian - dentro do ambiente isolado de construção, também chamdo de jaula (*chroot*).

Uma vez que a distribuição escolhida tenha sido o Debian GNU/Linux, este documento assumirá os comandos do mesmo para a manipulação de programas, a partir deste ponto.

4 Arquiteturas suportadas

O GabiX é capaz de gerar sistemas de 32 bits, 64 bits e, de forma muito experimental, ARM (armel). No caso do ARM, é necessário o uso de ferramentas adicionais, como plataformas de desenvolvimento específicas para a CPU escolhida ou máquinas virtuais que simulem ARM, a exemplo o *Q-Emu*.

Para escolher qual arquitetura utilizar, basta usar o comando "make" para que o GabiX leia as configurações em seu "Makefile". A configuração de arquiteturas funciona da seguinte maneira:

```
gabix_x86
gabix_x86_64
gabix_armel
```

Estes são os diretórios que contém os arquivos que darão origem à imagem ISO final, de acordo com a arquitetura selecionada.

```
miniroot_x86
miniroot_x86_64
miniroot_armel
```

Estes são os diretórios que contém os arquivos que darão origem ao "initrd" da imagem ISO final, de acordo com a arquitetura selecionada.

4.1 Ajustar a configuração para 32 bits

Basta executar os comandos (considerando que a jaula esteja no diretório root-NFS-32bits).

```
make config_32
ln -s root-NFS-32bits root-NFS
```

4.2 Ajustar a configuração para 64 bits

Basta executar os comandos (considerando que a jaula esteja no diretório root-NFS-64bits).

```
make config_64
ln -s root-NFS-64bits root-NFS
```

4.3 Ajustar a configuração para ARM (armel)

Basta executar os comandos (considerando que a jaula esteja no diretório root-NFS-armel).

```
make config_armel
ln -s root-NFS-armel root-NFS
```

Assim, de acordo com a arquitetura selecionada, um link simbólico será criado para o respectivo par de diretórios "gabix" e "miniroot". A título de exemplo, é apresentado o *layout* do diretório de desenvolvimento para o caso de uso de cada arquitetura.

```
32 bits:
gabix -> gabix_x86
miniroot -> miniroot_x86
root-NFS -> root-NFS-32bits

64 bits:
gabix -> gabix_x86_64
miniroot -> miniroot_x86_64
root-NFS -> root-NFS-64bits

ARM (armel):
gabix -> gabix_armel
miniroot -> miniroot_armel
root-NFS -> root-NFS-armel
```

5 Makefile e a construção do GabiX

A construção do GabiX é feita a partir da execução, em uma sequência lógica, de algumas ferramentas escritas, inicialmente, em *Shell Script*.

Para facilitar a construção parcial ou total do GabiX, um Makefile pode ser utilizado como um adicional, aumentando a facilidade na montagem do sistema final. O uso desse arquivo reduz a chance de erros, uma vez que tem-se a garantia de sempre se executar a sequencia correta dos comandos envolvidos no processo de construção.

5.1 Makefile usado pelo GabiX

O Makefile do GabiX é muito documentado e fácil de entender. Abaixo, segue o código deste arquivo, na íntegra.

```
# Makefile para gerar o sistema GabiX
#
# Gabriel Marques
# snortt@gmail.com
#
# Qui Dez  8 15:38:47 BRST 2011
#

# Ajuda
help:
@-echo -e "\
config_32       - Ajusta o GabiX para 32 bits.\n\
config_64       - Ajusta o GabiX para 64 bits.\n\
config_armel    - Ajusta o GabiX para ARM (armel).\n\
all             - Passa por todas as etapas.\n\
clean           - Apaga os arquivos criados. ISO e SquashFS (mantém ajustes).\n\
clean_links     - Apaga apenas os links simbólicos criados (reset arquitetura).\n\
all_clean_soft  - Passa por todas as etapas.\n\
                  Remove arquivos de zonas não usadas e idiomas não usados.\n\
all_clean_hard  - Passa por todas as etapas.\n\
                  Remove arquivos de zonas não usadas, idiomas não usados e documentação.\n\
criar_miniroot  - Cria o initrd do GabiX.\n\
criar_iso_file  - Cria apenas o arquivo ISO.\n\
criar_squash_fs - Cria a imagem SquashFS do sistema."


# Regras para escolher entre 32 ou 64 bits
config_32:
@make --silent clean_links
@echo -e "Ajustando sistema para 32 bits"
@echo -e "------------------------------"
@-ln -s kernel/nvidia_drv/nv_live_x86 nv_live && echo -e "nv_live -> nv_live_x86 : [OK]"
@-ln -s gabix_x86 gabix && echo -e "gabix -> gabix_x86 : [OK]"
@-ln -s miniroot_x86 miniroot && echo -e "miniroot -> miniroot_x86 : [OK]"
@echo -e "Não esqueça de criar o link simbólico root-NFS para o seu sistema\n"

config_64:
@make --silent clean_links
@echo -e "Ajustando sistema para 64 bits"
@echo -e "------------------------------"
@-ln -s kernel/nvidia_drv/nv_live_x86_64 nv_live && echo -e "nv_live -> nv_live_x86_64 : [OK]"
@-ln -s gabix_x86_64 gabix && echo -e "gabix -> gabix_x86_64 : [OK]"
```

```
@-ln -s miniroot_x86_64 miniroot && echo -e "miniroot -> miniroot_x86_64 : [OK]"
@echo -e "Não esqueça de criar o link simbólico root-NFS para o seu sistema\n"

config_armel:
@make --silent clean_links
@echo -e "Ajustando sistema para ARM (armel)"
@echo -e "-----------------------------------"
@-ln -s gabix_armel gabix && echo -e "gabix -> gabix_armel : [OK]"
@-ln -s miniroot_armel miniroot && echo -e "miniroot -> miniroot_armel : [OK]"
@echo -e "Não esqueça de criar o link simbólico root-NFS para o seu sistema\n"

# Regra para criar o miniroot
#criar_miniroot: criar_miniroot.sh
criar_miniroot: criar_miniroot.sh
@./criar_miniroot.sh miniroot

# Regra para criar a imagem squashfs
criar_squashfs: criar_squashfs.sh
@-./criar_squashfs.sh root-NFS/

# Regra para gerar a imagem ISO
criar_iso_file: make-iso.sh
@./make-iso.sh gabix

# Regra que remove os arquivos antes de criar os novos
clean:
@echo -e "Limpando arquivos existentes"
@echo -e "----------------------------"
@-rm *.iso 2> /dev/null && echo -e "ISO : [OK]"  || echo -e "ISO : [Existe?]"
@-rm gabix_*/base/sistema 2> /dev/null && echo -e "base/sistema : [OK]" || \
    echo -e "base/sistema : [Existe?]"

# Regra que remove o driver da Nvidia
clean_nvidia_drv:
@echo -e "Limpando drivers existentes"
@echo -e "----------------------------"
@-rm -rf gabix_*/drv/nv_live 2> /dev/null && echo -e "Nvidia : [OK]"  || \
    echo -e "Nvidia : [Existe?]"

# Regra que remove o arquivo de controle de arquiteturas
clean_arch: .arch_control
@echo -e "Limpando configuração de arquitetura"
@echo -e "------------------------------------"
@-echo -e "" > .arch_control && echo -e "Arquivo de controle : [OK]"  || \
    echo "Arquivo de controle : [Existe?]"

# Regra que remove apenas os links simbólicos
clean_links:
@echo -e "Limpando links existentes"
@echo -e "-------------------------"
@-rm gabix 2> /dev/null && echo -e "gabix : [OK]" || echo -e "gabix : [Existe?]"
@-rm miniroot 2> /dev/null && echo -e "miniroot : [OK]" || echo -e "miniroot : [Existe?]"
@-rm nv_live 2> /dev/null && echo -e "nv_live : [OK]" || echo -e "nv_live : [Existe?]"
@-rm root-NFS 2> /dev/null && echo -e "root-NFS : [OK]" || echo -e "root-NFS : [Existe?]"
@make --silent clean_arch
```

```
# Regra para limpar arquivos do sistema para poupar espaco
clean_sysfiles: clean_sysfiles.sh
@echo -e "Limpando arquivos temporarios ..."
@-find root-NFS/ -name "*~" -exec rm -fv {} \;
@./clean_sysfiles.sh root-NFS/

# Regra para limpar arquivos do sistema (+ /usr/share/doc/*) para poupar mais espaco
clean_sysfiles_full: clean_sysfiles.sh
@echo -e "Limpando arquivos temporarios ..."
@-find root-NFS/ -name "*~" -exec rm -fv {} \;
@./clean_sysfiles.sh root-NFS/ -f

# Regra para executar todas as etapas de criação do GabiX e gerar a ISO final.
#all: clean criar_miniroot criar_squashfs criar_iso_file
all:
@make --silent clean
@make --silent clean_nvidia_drv
@make --silent criar_miniroot
@make --silent criar_squashfs
@make --silent criar_iso_file

# Esta remove os arquivos locales e timezones da raiz da imagem squashfs, deixando documentação.
#all_clean_soft: clean clean_sysfiles criar_miniroot criar_squashfs criar_iso_file
all_clean_soft: clean_sysfiles all

# Esta remove os arquivos de documentação, locales e timezones da raiz da imagem squashfs.
#all_clean_hard: clean clean_sysfiles_full criar_miniroot criar_squashfs criar_iso_file
all_clean_hard: clean_sysfiles_full all
```

6 *Initrd*

O initrd é um arquivo que contém o que é também chamado *miniroot*. Trata-se de um pequeno sistema de arquivos que contém o mínimo necessário para que o sistema funcione até o ponto de se ter um prompt de comandos. Um pouco mais do que isso, o initrd é o principal responsável pelo preparo do ambiente que a distribuição utiliza para carregar.

A construção do initrd é feita a partir da configuração, compilação e manipulação da ferramenta *BusyBox*, um executável que se comporta como diversos programas do Linux, dependendo da forma como é invocado na linha de comandos. Uma vez instalado em seu devido lugar, ferramentas adicionais serão criadas para utilizar o *BusyBox*. Antes, porém, será necessária a instalação de ferramentas no sistema hospedeiro, utilizadas para compilação de programas.

6.1 Preparo do ambiente para construir o *BusyBox*

O sistema hospedeiro precisa ser ajustado para ser capaz de compilar programas. Os pacotes instalados em seguida permitirão a compilação do *BusyBox* e do núcleo Linux utilizado pelo GabiX.

```
apt-get install make gcc binutils bin86 libncurses-dev \
                kernel-package vim ctags g++ automake
```

Em seguida, crie uma área para o desenvolvimento do GabiX. Este documento assumirá /GabiX-devel como sendo esta área. Diretórios adicionais também devem ser criados dentro da área de desenvolvimento. Estes são:

- gabix: Diretório que dá origem à imagem final do sistema (ex: gabix.iso)
- base: Diretório que contém os arquivos de sistema, em *squashfs*, dentro do diretório gabix
- kernel: Diretório que contém os arquivos fonte do kernel

```
mkdir -p /GabiX-devel/{gabix/base,kernel}
```

6.2 Como obter, configurar e instalar o *BusyBox*

A ferramenta pode ser obtida através da página oficial do projeto.

```
wget -c http://www.busybox.net/downloads/busybox-1.18.4.tar.bz2
```

Após baixar o código-fonte, extraia o conteúdo do arquivo para algum diretório e inicie a configuração e depois compile e instale o *BusyBox*.

```
tar xjf busybox-1.18.4.tar.bz2 && cd busybox-1.18.4
make menuconfig && make && make install
```

Terminada a sequência de comandos acima, a ferramenta estará disponível no diretório *_install*. É a partir deste diretório que o *miniroot* será construído. Copie o diretório *_install* para o local de desenvolvimento.

```
cp -ar _install /GabiX-devel/miniroot
```

Remova o arquivo *linuxrc*, que aponta para o *busybox*, uma vez que um novo *init* será criado para o sistema.

```
rm /GabiX-devel/miniroot/linuxrc
```

6.3 Diretório */dev*

O diretório */dev*, dentro do miniroot, é o responsável pelos dispositivos utilizados durante o carregamento do sistema, até o ponto em que a distribuição Linux esteja carregada e assuma o controle, com seu próprio diretório de dispositivos.

Este diretório pode variar em conteúdo de acordo com a aplicação do GabiX. A seguir, um exemplo de um diretório */dev* completo para o miniroot:

```
root@GabiX # ls miniroot/dev/

admmidi1 adsp amidi1 audio autofs block bsg bus cdrom cdrom1 cdrw cdrw1
char cloop cloop0 cloop1 cloop2 cloop3 cloop4 cloop5 cloop6 cloop7 console
core cpu cpu_dma_latency disk dmmidi1 dri dsp dvd dvd1 dvdrw dvdrw1 fb0 fd
fd0 full fuse hidraw0 hidraw1 hpet hwrng initctl input kmem kmsg log loop0
loop1 loop2 loop3 loop4 loop5 loop6 loop7 lp0 MAKEDEV mapper mcelog mem
midi1 mixer mixer1 net network_latency network_throughput null nvidia0
nvidiactl nvram oldmem parport0 port ppp psaux ptmx pts ram random rfkill
root rtc rtc0 scd0 scd1 scd2 scd3 scd4 scd5 scd6 scd7 sda sda1 sda2 sda3
sda4 sda5 sda6 sda7 sdb sdb1 sdb2 sdb3 sdb4 sdb5 sdb6 sdb7 sdc sdc1 sdc2
sdc3 sdc4 sdc5 sdc6 sdc7 sequencer sequencer2 sg0 sg1 shm snapshot snd
sndstat sr0 sr1 sr2 sr3 sr4 sr5 sr6 sr7 stderr stdin stdout tty tty0 tty1
tty10 tty11 tty12 tty13 tty14 tty15 tty16 tty17 tty18 tty19 tty2 tty20 tty21
tty22 tty23 tty24 tty25 tty26 tty27 tty28 tty29 tty3 tty30 tty31 tty32 tty33
tty34 tty35 tty36 tty37 tty38 tty39 tty4 tty40 tty41 tty42 tty43 tty44 tty45
tty46 tty47 tty48 tty49 tty5 tty50 tty51 tty52 tty53 tty54 tty55 tty56 tty57
tty58 tty59 tty6 tty60 tty61 tty62 tty63 tty7 tty8 tty9 ttyS0 ttyS1 ttyS2
ttyS3 uinput urandom usb usbdev1.1 usbdev2.1 usbdev3.1 usbdev3.2 usbdev4.1
usbdev5.1 usbmon0 usbmon1 usbmon2 usbmon3 usbmon4 usbmon5 vcs vcs1 vcs2 vcs3
vcs4 vcs5 vcs6 vcs7 vcs8 vcsa vcsa1 vcsa2 vcsa3 vcsa4 vcsa5 vcsa6 vcsa7 vcsa8
vga_arbiter vhci xconsole zero
```

Seu conteúdo pode ser gerado de forma manual, através de consultas ao arquivo devices.txt do kernel ou a partir de uma cópia de arquivos de um diretório de dispositivos já existente.

6.3.1 Criação manual dos dispositivos

O diretório de fontes do kernel Linux contém um arquivo, *Documentation/devices.txt*, que lista todos os dispositivos e suas características. Com ele é possível utilizar a ferramenta *mknod* para criar dispositivos do sistema, desde que, quando apropriado, haja o hardware a ser utilizado por cada um deles.

A exemplo, tem-se os discos rígidos do tipo SATA, representados pelas iniciais *sd*, seguidas da letra referente à sua porta de conexão (a = SATA1, b = SATA2, ...). Discos rígidos são dispositivos de bloco. Além disso, dois números auxiliam na identificação, o número maior (*major*) e o número menor *minor*. No caso de um disco, o número maior representa sua posição na placa. O menor indica suas partições. O documento *devices.txt* diz que um disco SATA, estando na porta 1, tem número maior 8 e número menor 0. A primeira partição desse disco terá número maior 8 e número menor 1 e assim por diante, até o limite do número menor, 15, quando o dispositivo representado muda para um disco SATA na porta 2. Por sua vez, o segundo disco SATA terá seu número maior 8 e o menor variando entre 16 e 31, quando o dispositivo mudará para o terceiro disco SATA e seu número menor começará em 32.

Juntando as informações fornecidas nas linhas acima, pode-se criar um arquivo sda, que é do tipo bloco e representa um disco SATA, ligado à primeira porta da placa-mãe. Também é possível criar arquivos que representem suas partições. O mesmo pode ser repetido para os discos nas demais portas SATA.

```
mknod miniroot/dev/sda b 8 0
mknod miniroot/dev/sda1 b 8 1
mknod miniroot/dev/sda2 b 8 2
mknod miniroot/dev/sdb b 8 16
mknod miniroot/dev/sdb1 b 8 17
mknod miniroot/dev/sdb2 b 8 18
```

Dessa forma, seguindo as informações encontradas no arquivo de dispositivos, é possível criar, manualmente, todos os dispositivos que o sistema precisará para carregar.

6.3.2 Criação de cópia dos dispositivos

Uma maneira mais rápida (e menos propícia a erros) de criar dispositivos no miniroot é a simples cópia dos arquivos já existentes no diretório */dev* do sistema hospedeiro para dentro do miniroot. Vale notar que a

cópia de arquivos de disposivos requer a opção *-a, –archive* do comando *cp*, que preserva as características do arquivo copiado. O exemplo a seguir ilustra como copiar o diretório */dev* do sistema hospedeiro para o miniroot.

```
cp -ar /dev /GabiX-devel/miniroot/
```

Uma vez que o busybox tenha sido instalado e o diretório *dev/* criado, diretórios adicionais devem ser criados para que o miniroot tenha uma estrutura semelhante à ilustrada abaixo:

```
miniroot/
|-- bin
|-- cdrom
|-- dev
|-- dynamic
|-- etc
|-- lib
|   |-- modules
|
|-- mnt
|-- proc
|-- root
|-- sbin
|-- static
|-- sys
|-- tmp
|-- union
|-- usr
|    |-- bin
|    |-- sbin
|
|-- var
     |-- lock
```

6.4 init e funcoes.rc

Responsável pelas principais funções de preparo do ambiente para a distribuição Linux carregar, o init é o script que fica na raiz do miniroot e, após suas operações, deve passar o controle para o init verdadeiro, localizado no diretório */sbin/* da distribuição.

No GabiX, o init não trabalha sozinho. Ele precisa de um outro arquivo, funcoes.rc, que contém as principais funções utilizadas para tratar parâmetros de linha de comando do kernel, localizar a unidade de CD/DVD-ROM, montar dispositivos e, principalmente, preparar o ambiente a ser passado para a distribuição Linux que será carregada a partir da mídia.

O *script init* deve estar na raiz do miniroot. Atente para o fato de que o sistema utiliza *templates* de arquivos. Este arquivo é criado a partir do *template* localizado em "confs/boot/grub,armel".

A seguir está a listagem, na íntegra, deste script no GabiX:

```
#!/bin/sh
#
# Init para miniroot baseado em busybox
#
# Gabriel Marques
# snortt@gmail.com
#
# Qua Abr  6 10:02:17 BRT 2011
#

# Carrega o arquivo de configuracoes
```

```
if [ -f /etc/funcoes.rc ]; then
    . /etc/funcoes.rc
fi

#
# ------------------------------------------------------------------------
# Aqui comeca a magica
# --------------------
# Alo, pessoal!
printMsg " "
printMsg "${ADERECO}-------------------------------------------------"
printMsg "${FORTE}*** ${ACERTO}Bem-vindo ao ${AVISO}GabiX-LiveMaker ${FORTE}***"
printMsg "${ADERECO}-------------------------------------------------"

printMsg " "
printMsg 1 "Criando sistema de arquivos para dispositivos"
(
mount -t tmpfs -o size=64k,mode=0755 tmpfs /dev
mkdir /dev/pts
mount -t devpts devpts /dev/pts
) && printOK || printError

printMsg " "
printMsg 1 "Criando nodes especiais"
mknod -m 660 /dev/console c 5 1 && printOK "${MAGENTA}Console :: " || printError
mknod -m 660 /dev/null c 1 3 && printOK "${MAGENTA}Null :: " || printError

printMsg " "
printMsg 1 "Sistemas de arquivos do kernel"
(mount -t proc none /proc && printOK "${MAGENTA}proc") || printError
(mount -t sysfs none /sys && printOK "${MAGENTA}sys") || printError

printMsg " "
printMsg 1 "Habilitando mdev"
(echo -e  "/sbin/mdev" > /proc/sys/kernel/hotplug && \
printOK "${MAGENTA}kernel hotplug :: ") || printError

printMsg " "
printMsg 1 "Criando dispositivos"
(mdev -s && printOK "${MAGENTA}Dispositivos" || printError)

printMsg " "
printMsg 1 "Criando lock"
(mkdir -p /var/lock && printOK "${MAGENTA}lock" ) || printError

printMsg " "
printMsg 1 "Criando dependencias"
(depmod && printOK "${MAGENTA}Dependencias" || printError)

# Armazena a linha de comando passada para o kernel
CMDLINE=$(cat /proc/cmdline)

# Caso precise de uma porta serial
#printMsg "Ajustando porta serial"
#(stty -F /dev/ttyS0 -raw -echo 38400 && printOK) || printError
```

```
printMsg " "
printMsg 1 "Reduzindo ruidos do kernel"
(echo 0 > /proc/sys/kernel/printk && printOK) || printError

# Carrega modulos do kernel
load_modules $MODULES

printMsg " "
printMsg 1 "Procurando unidade de CD-ROM"
find_cdrom

# Macete feio, para ficar mais bonito! ;-)
printMsg " "

# Daqui pra frente, uma vez que o CD-ROM tenha sido localizado e montado,
# qualquer erro sera fatal para o processo de carga do sistema final.
# Assim, qualquer problema escreve 1 no arquivo de erro e faz com que o
# sistema jogue a inicializacao pela janela e carregue um shell.
#
if [ "$found_cd" -eq 0 ]; then
printError "${CIANO}CD-ROM :: "
echo "1" > $ERRFILE
fi

# Aqui precisamos verificar se foi solicitado um BOOT em raiz alternativa (local)
check_root

# E tambem se foi solicitado um shell
# Vamos pedir também para que seja feita a montagem das imagens squashfs.
check_shell "monta_squashes"

#
# Caso um shell não tenha sido solicitado, continuamos a carregar o sistema ...
#
# Vamos montar uma area dinamica para garantir escrita no sistema de arquivos
printMsg " "
printMsg 1 "Montando area dinamica"
(mount -t tmpfs tmpfs /dynamic > /dev/null 2>&1 && printOK) || \
(echo "1" > $ERRFILE && printError)

# Realiza as montagens das imagens SQUASHFS
montar_squashes

# Caso um HOME local tenha sido especificado, use-o no sistema live
check_home "static"

# Agora vamos unir a area estatica (RO) com a dinamica (RW)
unir_tudo

# Qual o tipo de configuracao de rede foi solicitado na linha de inicializacao?
check_ip_config

# Tenta executar o sistema real.
if grep 0 $ERRFILE > /dev/null 2>&1; then
```

```
    # So moveremos o proc e sys quando nao houver erros
    printMsg " "
    printMsg 1 "Movendo sistemas de arquivos virtuais"
    move_fs "/sys" "/union/sys"
    move_fs "/proc" "/union/proc"
move_fs "/dev" "/union/dev"

    # Finalmente, tenta trocar a raiz para o novo sistema e executar o init real.
    printMsg " "
    printMsg 1 "${AVISO}Tentando trocar raiz de lugar"
    printMsg " "
    printMsg 1 "${ACERTO}Iniciando"
    exec switch_root -c /dev/console /union /sbin/init 2> /dev/null
else
    printMsg " "
    printMsg 1 "${ERRO}Erro ao tentar iniciar Sistema!"
printMsg 1 "${ERRO}Executando Shell interativo. :-("
    exec_shell
fi
```

O *script funcoes.rc* deve estar na raiz do miniroot, dentro do diretório *etc/*. A seguir está a listagem, na íntegra, deste script no GabiX:

```
#!/bin/sh
#
# Funcoes utilizadas pelo miniroot baseado em busybox
#
# Gabriel Marques
# snortt@gmail.com
#
# Seg Abr 18 10:37:01 BRT 2011
#

# Modulos iniciais a serem carregados
MODULES="loop squashfs unionfs ext3 ext4 xfs"

# Possiveis candidatos a unidade de CD-ROM
#CDROMDEV="hda hdb hdc hdd hde hdf hdg hdh sda sdb sdc sdd sde \
#          sdf sdg sdh sr0 sr1 sr2 sr3"
CDROMDEV="sda sdb sdc sdd sde sdf sdg sdh sr0 sr1 sr2 sr3"

# raiz do cdrom
cdromroot="/cdrom"
# base na raiz do cdrom
cdrombase="${cdromroot}/base"
# Arquivo que contem o sistema real, em squashfs
base="sistema"
# Arquivo que contem os modulos adicionais do kernel, em squashfs
mods="modulos"
# Arquivo que contem firmware utilizados pelo kernel, em squashfs
fmwr="firmware"
# Arquivo que contem headers utilizados pelo kernel, em squashfs
khdr="headers"

# --------------------------
# Diretórios na área estática. Para facilitar nos testes e no uso geral.
```

```
STATIC_BASE="/static"
STATIC_MODS="${STATIC_BASE}/lib/modules/"
STATIC_HDR="${STATIC_BASE}/usr/src/"
STATIC_FIRMWARE="${STATIC_BASE}/lib/firmware/"
# ---------------------------
# A mesma coisa, porém para local alternativo.
# Algum linux já instalado localmente.
ALTER_STATIC_BASE="/mnt"
ALTER_STATIC_MODS="${ALTER_STATIC_BASE}/lib/modules/"
ALTER_STATIC_HDR="${ALTER_STATIC_BASE}/usr/src/"
ALTER_STATIC_FIRMWARE="${ALTER_STATIC_BASE}/lib/firmware/"

# Arquivo de controle de erros
# Precisamos dele porque os exports de subshells nao sobrescrevem
# os de nivel anterior
# ERRFILE = 0 : nao houve erro
# ERRFILE = 1 : houve erro
ERRFILE="/tmp/error.tmp"
echo "0" > $ERRFILE

# Carrega o sistema de cores
# É importante que você leia o arquivo de cores para entender os
# padrões utilizados nas mensagens.
# Isso também vai facilitar sua vida, se desejar trocar as cores do sistema.
if [ -f /etc/cores.defs ]; then
    . /etc/cores.defs
fi

# Esta funcao ‘parseia‘ a linha de comando de boot do kernel
# $1 e o valor que voce procura
# $2 e a variavel que sera inicializada com 1, caso $1 seja encontrado
# Se $1 for "root", "restore" ou "home", esta funcao procura pelo dispositivo
# especificado e exporta uma variavel de acordo.
#
# Isso é utilizado para manipular as operacoes de boot:
# * Inicializar em sistema raiz alternativo
# * Definir o home do usuario em outro local (um disco, talvez?)
# * Recuperar configuracoes previamente armazenadas em um arquivo
# * Inicializar um linux local
# * Determinar configuração de IP
# * Solicitar carga de módulos adicionais
#
parse_cmd_line() {
  for field in $CMDLINE
  do
    if echo "$field" | grep "${1}" >/dev/null 2>&1
    then
      # Encontramos um pedido de var= na linha de comandos?
      # Se sim, entao salve-o em uma variavel!
        case "$1" in
           "root") export ROOTDEV=$(echo $field | \
                awk -F "=" '{print $2}') ;;

           "restore") export RESTOREDEV=$(echo $field | \
                awk -F "=" '{print $2}') ;;
```

```
            "home") export HOMEDEV=$(echo $field | \
                 awk -F "=" '{print $2}') ;;

            "ip_config") export IP_CONFIG=$(echo ${CMDLINE#*ip_config} | \
                 cut -d "\"" -f2 | cut -d " " -f1)

                # Se o IP for estatico, vamos ver se foi passado.
                if [ "$IP_CONFIG" == "static" ]; then
                    export IP_STATIC=$(echo ${CMDLINE#*ip_config} | \
                             cut -d "\"" -f2 | cut -d " " -f2)

                    export NET_MASK=$(echo ${CMDLINE#*ip_config} | \
                         cut -d "\"" -f2 | cut -d " " -f3)

                    export NET_DNS=$(echo ${CMDLINE#*ip_config} | \
                         cut -d "\"" -f2 | cut -d " " -f4)

                    export DFGW=$(echo ${CMDLINE#*ip_config} | \
                         cut -d "\"" -f2 | cut -d " " -f5)

                    # Se o usuario pular algum dos valores, é óbvio que a configuração
                    # da rede vai ficar toda detonada com os valores abaixo.
                    # Mas ainda assim, vamos usar valores comuns para
                    # evitar problemas na inicialização do Debian.
                    if [ -z "$IP_STATIC" ]; then export IP_STATIC="10.0.0.1"; fi
                    if [ -z "$NET_MASK" ]; then export NET_MASK="255.0.0.0"; fi
                    if [ -z "$NET_DNS"  ]; then export NET_DNS="10.0.0.100"; fi
                    if [ -z "$DFGW" ]; then export DFGW="10.0.0.254"; fi
                fi
                ;;

            # Lembre-se de que modulos="m1 m2 ...".
            "modulos") export MODS_CONFIG=$(echo ${CMDLINE#*modulos} | \
                 cut -d "\"" -f2) ;;

        esac
        export ${2}="1"
        break
    else
        export ${2}="0"
    fi
  done
}

# Funcao que executa um shell em um tty (e nao console)
exec_shell() {
    # Vamos deixar o nome GabiX, pois é a base de toda essa parafernalha, né? ;-)
    export PS1="${VERDE}GabiX${AMARELO}# ${RESET}"
    exec setsid sh -c 'exec sh </dev/tty1 >/dev/tty1 2>&1'
}

# Funcao que imprime mensagens de sucesso
printOK() {
    echo -e "$@ ${ADERECO}[${ACERTO}OK${ADERECO}]${RESET}"
```

```
}

# Funcao que imprime mensagens de erro
printError() {
    echo -e "$@ ${ADERECO}[${ERRO}Erro${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens de aviso
printWarn() {
    echo -e "$@ ${ADERECO}[${AVISO}Aviso${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens
# Se receber 1 em $1, imprime mensagem extendida ("*** TEXTO ***").
# Do contrário, imprime mensagem comum ("TEXTO").
printMsg() {
    if [ "$1" == "1" ]; then
        shift
        echo -e "${ADERECO}*** ${TEXTO}$@ ${ADERECO}***${RESET}"
    else
        echo -e "${TEXTO}$@ ${RESET}"
    fi
}

# Funcao que carrega os modulos do kernel
load_modules() {
    printMsg " "
    printMsg 1 "Carregando modulos"
    for module in $@
    do
        printMsg "modulo ${ADERECO} :: ${MOD}$module"
     (modprobe $module && printOK) || (echo "1" > $ERRFILE && printError)
    done

    # Vamos ver se módulos adicionais foram solicitados.
    parse_cmd_line "modulos" MODS_CONFIG_ON
    if [ "$MODS_CONFIG_ON" -eq "1" ]; then
        # Arquivo de log
        modprobe_log_file="/tmp/modprobe.log"

        printMsg " "
        printMsg 1 "Carregando modulos adicionais"
        for modulo in $MODS_CONFIG
        do
            modprobe $modulo &> $modprobe_log_file
            file_size=$(stat -c %s $modprobe_log_file)
            if [ "$file_size" -eq "0" ]; then
                printOK "${ADERECO} modulo : ${MOD} $modulo "
            else
                printError "${ADERECO} modulo : ${MOD} $modulo "
            fi
        done
        # Vamos limpar a bagunça
        rm -f $modprobe_log_file
    fi
```

```
}

# Funcao para veriricar se foi solicitado um shell
# Se receber a string "monta_squashes", ela monta as imagens squashfs antes
# de executar o shell
check_shell() {
    parse_cmd_line "shell" SHELL
    if [ "$SHELL" -eq "1" ]
    then
        if [ "$1" == "monta_squashes" ]; then
            montar_squashes
        fi

        printMsg " "
        printMsg "${EXEC_SHELL}Executando Shell:"
        printMsg " "
        exec_shell
    fi
}

# Funcao para localizar um CD-ROM
find_cdrom() {
#   echo -en "${SEPARADOR}|${RESET}"
#    for cdrom in $CDROMDEV
#    do
     echo -en "${DEV}$cdrom${SEPARADOR}|${RESET}"
#     if mount -t iso9660 /dev/${cdrom} ${cdromroot} > /dev/null 2>&1; then
     if mount -t iso9660 ${CDROMDEV} ${cdromroot}; > /dev/null 2>&1; then
     #printOK "\n${ADERECO}CD-ROM :: ${DEV}$cdrom ${ADERECO} :: "
     printOK "\n${ADERECO}CD-ROM :: ${CDROMDEV} ${ADERECO} :: "
     export found_cd="1"
     echo "0" > $ERRFILE
     break
     fi
#    done
}

# Funcao para montagem dos sistemas de arquivos
# $1 e o dispositivo
# $2 e o ponto de montagem
mountit() {
    printMsg " "
    printMsg 1 "Montando ${DEV}${1}"
    (mount $1 $2 > /dev/null 2>&1 && printOK "${DEV}$1 ${ADERECO}:: ") || \
        (echo "1" > $ERRFILE && printError)
}

# Funcao para mover os sistemas de arquivos para outro local
# $1 representa o fs original
# $2 respresenta o novo ponto de montagem
move_fs() {
    printMsg 1 "Movendo ${DEV}${1}"
    (mount -n -o move $1 $2 > /dev/null 2>&1 && printOK "${DEV}$1 \
        ${APONTADOR}-> ${DEV}${2} ${ADERECO}:: ") \
        || (echo "1" > $ERRFILE && printError)
```

```
}

# Funcao para montar um home alternativo
# $1 define que o home sera montado no sistema do CD-ROM se for "static"
# $1 define que o home sera montado no sistema local se for "mnt"
#
check_home() {
    # Verifica se tambem foi solicitado um HOME local
    parse_cmd_line "home" "HOME_LOCAL"
    if [ "$HOME_LOCAL" -eq "1" ]
    then
        printMsg " "
        printMsg 1 "Usando HOME local em ${DEV}${HOMEDEV}"
        mountit "$HOMEDEV" "/home"
        if grep 0 $ERRFILE > /dev/null 2>&1; then
            # Caso não exista um 'home' no local especificado,
            # vamos criar o nosso!
            # Assim, as configs do usuário já ficarão lá também ;-)
        if [ ! -d ${1}/home ]; then
                printMsg " "
                printMsg "${AVISO}${1}/home nao encontrado em \
                    ${DEV}${HOMEDEV}. Criando um ..."
         (mkdir -p ${1}/home && printOK) || \
             (echo "1" > $ERRFILE && printError)
        fi
        move_fs "/home" "${1}/home"
# Se o 'home' der errado, que tal executar um shell?
#else
#exec_shell
    fi
    fi
}

# Funcao para veriricar se o IP deve ser estatico ou dinamico
# Caso o usuario escolha IP estatico, IP_CONFIG sera static
# Se ele optar por IP dinamico, IP_CONFIG sera dhcp
# Se ele apagar a variavel ip_config da linha de comando do kernel,
# IP_CONFIG_ON
# sera 0 e nao teremos configuracoes de rede.
check_ip_config() {
    parse_cmd_line "ip_config" "IP_CONFIG_ON"
    if [ "$IP_CONFIG_ON" -eq "0" ]
    then
        printMsg " "
        printMsg "${AVISO}Sem configuracao de rede"
    else
        if [ "$IP_CONFIG" == "static" ]
        then
            printMsg " "
            printMsg "Usando IP ${ADERECO}[${AVISO}estatico${ADERECO}]"
            # Aqui o erro nao nos importa mais (ja vimos que o recurso funciona)
            #cp /etc/interfaces.static /union/etc/network/interfaces 2> /dev/null
            cat /etc/interfaces.static | sed s/"_IP_STATIC_/${IP_STATIC}"/g | \
                sed s/"_NET_MASK_/${NET_MASK}"/g > \
                /union/etc/network/interfaces 2> /dev/null
```

```
        elif [ "$IP_CONFIG" == "dhcp" ]
        then
            printMsg " "
            printMsg "Usando IP ${ADERECO}[${AVISO}dinamico${ADERECO}]"
            # Aqui o erro nao nos importa mais (ja vimos que o recurso funciona)
            cp /etc/interfaces.dhcp /union/etc/network/interfaces 2> /dev/null
        fi
    fi
}

# Funcao para montar as imagens squashfs dos arquivos usados pelo kernel
# $1 e a origem
# $ e o ponto de montagem
mountit_kernel_stuff() {
    (mount -t squashfs "$1" "$2" -o loop > /dev/null 2>&1 && printOK) || \
        (echo "1" > $ERRFILE && printError)
}

# Funcao para verificar se foi solicitada uma raiz alternativa na
# linha de comandos
check_root() {
    parse_cmd_line "root" "ROOT_LOCAL"
    if [ "$ROOT_LOCAL" -eq "1" ]
    then
        # Tenta montar o dispositivo raiz
        printMsg " "
        printMsg 1 "Solicitado BOOT com raiz em ${DEV}${ROOTDEV}"
        mountit "${ROOTDEV}" "${ALTER_STATIC_BASE}"

        # Vamos ver se tambem foi solicitado um HOME alternativo
        check_home "${ALTER_STATIC_BASE}"

        # Uma vez que estamos usando o kernel Linux do GabiX, sera \
        # necessario usar os modulos do mesmo kernel
        # Vamos montar nossa imagem squashfs que contem os modulos
        # adicionais do kernel
        printMsg " "
        printMsg 1 "Montando imagem de modulos do kernel"
        mountit_kernel_stuff "${cdrombase}/${mods}" "${ALTER_STATIC_MODS}"

        # Vamos montar nossa imagem squashfs que contem o firmware
        # usado pelo kernel
        printMsg " "
        printMsg 1 "Montando imagem de firmware do kernel"
        mountit_kernel_stuff "${cdrombase}/${fmwr}" "${ALTER_STATIC_FIRMWARE}"

        # Vamos montar nossa imagem squashfs que contem os headers usados pelo kernel
        printMsg " "
        printMsg 1 "Montando imagem de headers do kernel"
        mountit_kernel_stuff "${cdrombase}/${khdr}" "${ALTER_STATIC_HDR}"

        # Antes de passar o controle, vamos ver se foi solicitado um shell.
        # Quem sabe? ;-)
        check_shell
```

```
        # Passa o controle para o sistema na raiz escolhida, caso nao
        # haja erros
        if grep 0 $ERRFILE > /dev/null 2>&1; then
            # Move os sistemas de arquivos virtuais para o novo
            # dispositivo raiz somente se tudo deu certo
            printMsg " "
            printMsg 1 "Movendo sistemas de arquivos virtuais para \
                ${DEV}${ROOTDEV}"
            move_fs "/sys" "${ALTER_STATIC_BASE}/sys"
            move_fs "/proc" "${ALTER_STATIC_BASE}/proc"

         #echo -e "\n${AZUL}*** ${CIANO}Executando sistema em \
               ${MAGENTA}${ROOTDEV} ${AZUL}***${RESET}"
            printMsg " "
            printMsg 1 "Executando sistema em ${DEV}${ROOTDEV}"
            printMsg " "
            printMsg 1 "${ACERTO}Iniciando"
            exec switch_root -c /dev/console ${ALTER_STATIC_BASE} "/sbin/init"
        else
            printMsg " "
            printMsg 1 "${ERRO}Houve um problema ao carregar o sistema. :-("
            printMsg " "
           exec_shell
        fi
  fi
}

# ---------------------------------------------------------
# Funções de encapsulamento das operações de inicialização.
# ---------------------------------------------------------
#
# Esta função utiliza as funções de montagem para encapsular as operações
# executadas durante o boot.
montar_squashes() {
    # Vamos montar o nosso sistema squashfs na area estatica
    printMsg " "
    printMsg 1 "Montando imagem squashfs do sistema"
    mountit_kernel_stuff "${cdrombase}/${base}" "$STATIC_BASE"

    # Aqui temos dois caminhos:
    # 1-) Sua imagem SQUASHFS do sistema foi encontrada e montada!
    #     Prossiga e monte as imagens dos módulos, firmware e headers
    # 2-) Sua imagem SQUASHFS do sistema não foi encontrada/montada!
    #     Ainda assim, vc pode querer montar as demais imagens.
    #     Assim, podemos incluir um teste para criar os diretórios
    #     ausentes pela não montagem da imagem SQUASHFS do sistema.
    #
    # Eu optei por seguir o segundo caminho.
    #
    # Vamos montar nossa imagem squashfs que contem os modulos
    # adicionais do kernel
    printMsg " "
    printMsg 1 "Montando imagem de modulos do kernel"
    if [ ! -d "$STATIC_MODS" ]; then
        printMsg "${ADERECO}Criando ${STATIC_MODS} nao funcional \
```

```
            [${ERRO}!${ADERECO}]"
        mkdir -p "$STATIC_MODS"
    fi
    # mountit_kernel_stuff "/cdrom/base/${mods}" "/static/lib/modules/"
    mountit_kernel_stuff "${cdrombase}/${mods}" "${STATIC_MODS}"

    # Vamos montar nossa imagem squashfs que contem o firmware usado pelo kernel
    printMsg " "
    printMsg 1 "Montando imagem de firmware do kernel"
    if [ ! -d "$STATIC_FIRMWARE" ]; then
        printMsg "${ADERECO}Criando ${STATIC_FIRMWARE} nao funcional \
            [${ERRO}!${ADERECO}]"
        mkdir -p "$STATIC_FIRMWARE"
    fi
    # mountit_kernel_stuff "/cdrom/base/${fmwr}" "/static/lib/firmware/"
    mountit_kernel_stuff "${cdrombase}/${fmwr}" "${STATIC_FIRMWARE}"

    # Vamos montar nossa imagem squashfs que contem os headers usados pelo kernel
    printMsg " "
    printMsg 1 "Montando imagem de headers do kernel"
    if [ ! -d "$STATIC_HDR" ]; then
        printMsg "${ADERECO}Criando ${STATIC_HDR} nao funcional \
            [${ERRO}!${ADERECO}]"
        mkdir -p "$STATIC_HDR"
    fi
    mountit_kernel_stuff "${cdrombase}/${khdr}" "${STATIC_HDR}"

    # Isso aqui permite que o BOOT fique visivel de dentro do sistema final
    # Precisa ser --bind porque /cdrom/boot nao e um pto de montagem a ser
    # realocado (mount --help, ;-P )
    printMsg " "
    printMsg 1 "Juntando arquivos do kernel ao sistema final"
    if [ ! -d "${STATIC_BASE}/boot" ]; then
        printMsg "${ADERECO}Criando ${STATIC_BASE}/boot nao funcional \
            [${ERRO}!${ADERECO}]"
        mkdir -p "$STATIC_BASE/boot"
    fi
    (mount -n --bind /cdrom/boot ${STATIC_BASE}/boot > /dev/null 2>&1 && printOK) \
        || (Erro="1" && printError)

    # Isso aqui permite que o CD-ROM fique visivel de dentro do sistema final
    printMsg " "
    printMsg 1 "Juntando CD-ROM ao sistema final"
    if [ ! -d "${STATIC_BASE}/cdrom" ]; then
        printMsg "${ADERECO}Criando ${STATIC_BASE}/cdrom nao funcional \
            [${ERRO}!${ADERECO}]"
        mkdir -p "$STATIC_BASE/cdrom"
    fi
    # Macete feio, para ficar bonito! ;-)
    # printMsg " "
    move_fs "${cdromroot}" "${STATIC_BASE}/cdrom"
}

# Função para encapsular a união das áreas estática e dinâmica
unir_tudo() {
```

```
    printMsg " "
    printMsg 1 "Unindo area dinamica e estatica"
    printMsg "${DEV}static ${AVISO}+ ${DEV}dynamic ${AVISO}= ${DEV}union"
    (mount -t unionfs unionfs /union -o dirs=/dynamic=rw:/static=ro > \
        /dev/null 2>&1 && printOK) || (Erro="1" && printError)
}
```

6.5 Cores ASCII

Para melhorar a visualização das tarefas executadas durante a inicialização do GabiX, um terceiro *script*, "cores.defs", lida com as cores ASCII através de variáveis fáceis de utilizar. O *script* em questão é apresentado, na íntegra, abaixo:

```
#!/bin/bash
#
# Este arquivo pode ser utilizado para gerenciar as cores do shell
# Gabriel Marques - snortt@gmail.com
#

# Cores ASCII
RESET='\033[0m'
BRANCO='\033[39;1m'
VERMELHO='\033[31;1m'
VERDE='\033[32;1m'
AMARELO='\033[33;1m'
AZUL='\033[34;1m'
MAGENTA='\033[35;1m'
CIANO='\033[36;1m'
BRANCOB='\033[37;1m'

# Cores usadas nas mensagens de texto ao longo do boot
# Se desejar criar temas, é aqui que você deve modificar.
TEXTO="$CIANO"
FORTE="$BRANCOB"
ADERECO="$AZUL"
DEV="$MAGENTA"
MOD="$MAGENTA"
ACERTO="$VERDE"
AVISO="$AMARELO"
ERRO="$VERMELHO"
SEPARADOR="$AMARELO"
APONTADOR="$VERMELHO"
EXEC_SHELL="$VERMELHO"
```

6.6 Criação do miniroot

Uma vez montado, o miniroot deve ser armazenado em um arquivo gerado através da junção do *cpio* e *gzip*, e chamar-se initrd-*versão*. Embora este processo possa ser realizado manualmente, é aconselhado o uso de um *script*, o que diminui a chance de erros e automatiza o procedimento.

6.6.1 Criação automática do miniroot

O miniroot pode ser gerado simplesmente digitando um dos comandos:

```
./criar_rootfs miniroot
make criar_miniroot
```

Uma vez que o miniroot esteja construído, é hora de configurar e ajustar o kernel Linux para o GabiX.

7 Sobre o kernel do Linux

Embora muita gente considere como sendo o sistema operacional tudo o que aparece na tela do computador, é importante ressaltar que, tecnicamente, o sistema operacional é a parte do sistema computacional que é responsável por administrar e gerir os recursos básicos do *hardware*. Isso inclui o Kernel, os *drivers* de dispositivos, o interpretador de comandos (Bash, no caso do Linux), o gerenciador de BOOT (GRUB, no caso do Linux) e algum sistema básico de arquivos e utilitários de sistema.
Quando se trata de Linux, o Kernel é um binário monolítico, porém com suporte ao carregamento dinâmico de extensões (módulos) em tempo de execução.
Mesmo seguindo a linha de desenvolvimento do kernel do Unix (ser monolítico), ele não precisa ser igual porque não deriva diretamente da árvore de fontes do Unix. O Linux é tipo um Unix, mas não é um Unix. Isso permite que ele possa incluir qualquer melhoria que ache prudente em seu kernel, incluindo a capacidade de carregamento dinâmico de binários ao seu kernel monolítico por natureza.

7.1 Suporte modular x suporte embutido

Quando se configura o kernel para uma determinada máquina, é importante ter em mente que é possível configurar suporte aos periféricos e aos dispositivos de duas maneiras:

- Suporte embutido: representado por "[*]" ou por "<*>", indica que o suporte ao dispositivo e/ou periférico foi configurado para ir diretamente dentro do binário do kernel. Assim, a partir do momento em que o kernel carregar na memória, o suporte configurado também estará disponível.

- Suporte modular: representado por "<M>", indica que o suporte ao dispositivo e/ou periférico foi configurado para ficar dentro de um binário separado, que pode ser carregado sob demanda, depois que o kernel iniciou e já está na memória. Configurações modulares otimizam o uso de recursos da máquina.

A esta altura duas coisas podem surgir na cabeça do leitor:
1-) qual a diferença entre [*] e <*> ?
2-) Como saber se deve-se configurar suporte modular ou embutido?
Esse tipo de questionamento é mais do que natural para quem nunca viu uma configuração de kernel em sua frente. Respondendo às questões anteriores:

1-) A diferença entre [*] e <*> é o fato de tal recurso "ser embutido" ou "poder ser embutido". Isso mesmo. Dentro do menu de configuração do kernel, há várias lacunas, representadas por [ ] ou < >. Todas as lacunas que forem na forma "[ ]", representam um recurso que só pode ser configurado embutido. Todas as lacunas que forem na forma "< >", representam um recurso que pode ou não ser embutido, ficando a critério do administrador. Isso deixa margem para a resposta à segunda pergunta.

2-) Compilar kernels é uma questão que envolve conhecimento do *hardware* da máquina, conhecimento sobre o sistema Linux e prática.
À medida em que o administrador for compilando kernels, ficará mais experiente. Em um primeiro momento, o menu de configuração parece um pouco hostil e complexo. Não se assuste! Ele é grande sim, mas não morde se o leitor usar o sistema de ajuda que já vem com o próprio kernel e também este documento.
É importante ter em mente os dois pensamentos abaixo, como uma abordagem inicial para decidir entre embutido ou módulo:

- Tudo aquilo que é fundamental para o funcionamento inicial do computador deve ser embutido.

- Tudo aquilo que pode ser removido sem que o computador perca sua funcionalidade, deve ser um módulo.

Disco rígido, memória, vídeo e *chipset* correto da placa-mãe são exemplos de componentes que impedem o computador de funcionar, se forem removidos. Esses devem sempre ser configurados como sendo recursos embutidos[7]. Tanto que alguns nem dão oportunidade de serem outra coisa que não seja embutido.
Unidade de CD-ROM, aceleração 3D, placa de som, placa de rede[8], unidade de disquetes, unidade de fita e mouse são exemplos de componentes que podem sr removidos sem que o computador deixe de ter sua

[7] A não ser que a pessoa seja um desenvolvedor e saiba o que está fazendo.
[8] Dependendo da aplicação da máquina

funcionalidade básica. Tais componentes são fortes candidatos a serem configurados como sendo módulos. Saber balancear entre embutido e módulo aumenta a eficiência do kernel e o desempenho da máquina de forma geral, até porque o sistema só alocará memória para o que ele realmente precisa, evitando desperdícios. Outra característica muito interessante dos módulos é que eles são dinâmicos! Eles podem ser carregados e descarregados, tanto pelo administrador, quanto pelo próprio sistema. Poder descarregar um módulo em tempo de execução é poder liberar memória em tempo de execução!

7.2 Onde obter o kernel mais novo

Para obter o kernel mais novo, o lugar mais apropriado para ir é o repositório oficial de kernels, na Internet. O sítio *http://www.kernel.org* é este local. Lá você encontra todas as versões do kernel do Linux, desde sua fase embrionária até os dias de hoje. Antes de ir para lá, é importante você compreender como é montado o nome do arquivo do kernel, pois é baseado nele que se saberá qual a versão deve obter.

7.3 Como saber a versão correta para baixar

Padrões de nomenclatura de distribuições à parte, o arquivo de fontes do kernel obtido no sítio oficial obedece à seguinte regra de nomenclatura:

linux-x.y.z.tar.bz2 Para arquivos compactados com bzip2

linux-x.y.z.tar.gz Para arquivos compactados com gzip

- x: número maior de versão.
- y: número menor de versão
- z: versão de lançamento

Os valores x.y compõem a versão atual do kernel. A exemplo tem-se os kernels 2.0, 2.1, 2.2, 2.3, 2.4, 2.5 e 2.6.
Sempre que o valor de y for um número par, o kernel será estável. Sempre que o valor de y for um número ímpar, o kernel será uma versão de desenvolvimento e, portanto, deverá ser evitado para nosso propósito.
Prefira sempre a versão 2.6. O kernel 2.5 é instável e o 2.7 está longe de surgir.

O terceiro valor, z, indica a versão de lançamento. O kernel muda de versão à medida que a versão de lançamento avança.
Prefira sempre o maior número de versão disponível, dentro da versão 2.6.

Alternativamente pode haver um quarto número, porém, deixemos isso para os desenvolvedores do kernel.
Prefira sempre a última versão, com o quarto número sendo par, caso haja um.
Se possível, escolha sempre o formato bzip2, que compacta melhor e suporta redundância de dados no arquivo.

7.4 Obtenção, extração, configuração e compilação

7.4.1 Pacotes necessários

Para configurar e compilar o kernel, é necessário ter alguns pacotes específicos no seu sistema. Instale-os com o comando:

```
apt-get install make gcc binutils bin86 libncurses-dev pciutils procps bzip2 kernel-package
```

7.4.2 Extração dos fontes para o local apropriado

O GabiX já vem com os fontes dos *kernels* que utiliza, disponíveis no diretório *kernel/src/*. Basta que o usuário vá ao diretório e extraia o kernel desejado.

```
cd kernel/src/3.3.1
tar xf linux-3.3.1.tar.bz2
```

7.5 Configuração do kernel sem um arquivo ".config" previamente criado

Para a etapa de configuração, há diversas alternativas. Pode-se editar o arquivo fonte diretamente, o que não é recomendado para iniciantes, pode-se usar uma interface de menu texto, o que também não é aconselhado por se tratar de um processo extremamente massante ou ainda, como a melhor alternativa, pode-se usar o menu baseado em "ncurses". Além disso, por se tratar de muitas opções, pode-se ainda usar o gerador automáico de configurações básicas. Trata-se de um comando que gera configurações padrão, baseadas na arquitetura do computador. Para isso, no diretório do kernel, o comando abaixo cria uma configuração básica:

```
cd kernel/src/3.3.1
make i386_defconfig
```

Uma vez que o GabiX faz uso de *patches* em seu *kernel*, é recomendada a leitura, mais abaixo, referente aos *patches* de que ele precisa, antes de prosseguir com a configuração realizada pela etapa seguinte. O comando a seguir inicia a interface do menu de configuração, usando a biblioteca "ncurses":

```
make menuconfig
```

Depois de configurado o kernel, as configurações devem ser salvas antes da saída do menu, o que é solicitado automaticamente ao administrador. Em seguida, o comando abaixo é utilizado para compilar e, se nenhuma configuração estiver errada, o kernel será gerado e empacotado em dois arquivos .deb (um para a imagem do kernel e outro para os arquivos de cabeçalhos).

```
make deb-pkg
```

Para sistemas multicore, a compilação paralela acelera a compilação. O argumento passado para o parâmetro *-j* representa o número de tarefas concorrentes (*jobs*) - sugere-se 2x número de núcleos da máquina.

```
make -j 8 deb-pkg
```

7.6 Configuração do kernel com arquivo ".config" previamente criado

Uma alternativa mais simples à de criar toda uma configuração completamente do zero seria a de aproveitar a configuração atual do kernel que vem com a distribuição e apenas compilar uma versão mais nova. A configuração do kernel em execução pode ser encontrada no arquivo */boot/config-x.y.z* (x.y.z dependendem da versão), ou ainda no arquivo */proc/config.gz*. De qualquer modo, é possível utilizar essas configurações como base e fazer as modificações adicionais, no menu de configuração do kernel.

7.6.1 Usando as configurações do arquivo do /proc

Se optar pelo arquivo do diretório proc, o administrador deve proceder da seguinte maneira:

```
cd kernel/src/3.3.1
zcat /proc/config.gz > .config
make oldconfig
```

Daqui pra frente, basta seguir os passos de configuração, compilação e instalação, apresentados nos itens anteriores.

7.6.2 Usando as configurações do arquivo da própria máquina

Se optar pelo arquivo do diretório /boot da própria máquina, o usuário deve proceder da seguinte maneira:

```
cd kernel/src/3.3.1
cat /boot/config-x.y.z > .config
make oldconfig
```

Daqui pra frente, basta seguir os passos de configuração, compilação e instalação, apresentados nos itens anteriores.

7.6.3 Usando as configurações do próprio GabiX

O GabiX já vem com seu arquivo de configuração do núcleo, disponível no diretório *gabix/boot/config-3.3.1-gabix*. Para usar este arquivo, o usuário deve copiá-lo para o seu diretório do kernel.

```
cd kernel/src/3.3.1/linux-3.3.1
cp ../../../../gabix/boot/config-3.3.1-gabix .config
make oldconfig
```

7.7 Gerenciamento de módulos

O Linux é capaz de carregar módulos sob demanda, mas nada impede o administrador de realizar operações sobre os módulos que selecionou para o seu kernel.
A ferramenta de gerenciamento de módulos é composta pelos comandos:

- modprobe: carrega um módulo e suas dependências.
- insmod: carrega um único módulo.
- rmmod: descarrega um único módulo.
- lsmod: lista os módulos carregados e suas dependências.
- modconf: ferramenta do Debian para gerenciamento de módulos, baseada em "ncurses".

Cada um desses comandos possui opções interessantes. Alguns exemplos:

```
modprobe -v modulo  <- carrega um módulo e suas dependências, mostrando na tela
modprobe -r modulo  <- remove um módulo da memória
insmod modulo opts  <- carrega um módulo e passa as opts para ele
rmmdo -f modulo     <- remove um módulo de forma forçada
lsmod | grep modulo <- lista todos os módulos e filtra o módulo desejado
```

Assim é possível carregar/descarregar suporte às unidades de mídias como CD-ROM, DVD-ROM, fitas, discos externos, cartões de memória, mouses, placas de rede, placas de som, etc.

8 *Kernel* Linux utilizado no GabiX

Uma das principais características do GabiX é o fato do sistema ter um *kernel* Linux configurado e compilado especialmente para ele. Este núcleo é montado a partir das configurações iniciais que acompanham o kernel utilizado pelo Debian GNU/Linux, porém com algumas modificações. O GabiX suporta, através de seu *kernel* Linux, recursos como:

- 64GB de memória RAM, utilizando PAE *Physical Address Extension*.
- 512 Núcleos de processamento em uma mesma máquina (CPU).
- Uma vasta gama de dispositivos de rede, bluetooth, wireless, etc.
- Praticamente todos os tipos de tabelas de partição suportados pelo Linux.
- Diversos sistemas de arquivos (ext2,3,4, XFS, ReiserFS, UnionFS, SquashFS, BtrFS, CIFS, Coda, AFS, etc.).
- O suporte à ferramenta *iptables* está com todos os módulos selecionados
- Sistemas de arquivos de rede
- Disposivitos de tecnologias de memória e sistemas de arquivos relacionados (NAND, NOR, YFFS2, etc.).
- Suporte para dispositivos de automação industrial (reguladores de voltagem, válvulas, etc.).

A lista de dispositivos e recursos suportados é bem completa e pode ser verificada através do arquivo de configuração do núcleo, */boot/config-VERSAO-gabix*, onde VERSAO é a versão atual do núcleo utilizado no sistema (2.6.38.8, na data em que este documento foi escrito).

8.1 *Patches utilizados no GabiX*

Abaixo são listados, acompanhados de uma breve descrição, os *patches* aplicados no núcleo Linux que acompanha o GabiX.

8.1.1 Suporte à transposição de sistemas de arquivos

Para que o suporte à transposição de sistemas de arquivos fosse possível, o núcleo utilizado no GabiX recebeu um patch para suportar o *UnionFS*, sistema de arquivos que permite montagem transparente entre sistemas de arquivos, mantendo visibilidade total do sistema sobreposto. O GabiX já vem com os fontes dos *patches* necessários.

A instalação do suporte para UnionFS foi realizada com a sequência de comandos:

```
cd kernel/src/3.3.1/linux-3.3.1
gunzip -c ../unionfs-2.5.11_for_3.3.0-rc3.diff.gz > unionfs-2.5.11_for_3.3.0-rc3.diff
patch -p1 < unionfs-2.5.11_for_3.3.0-rc3.diff
```

8.1.2 Patch para trocar o pinguim pelo pinguim com o logotipo do LNCC

O GabiX utiliza o pinguim clássico do Linux, porém com o logotipo do LNCC em seu peito. Para que isso fosse possível, algumas alterações foram realizadas nos fontes do kernel e, posteriormente, um *patch* foi gerado. Assim, basta que o administrador aplique o *patch* abaixo para que a opção de ligar o pinguim com o logotipo do LNCC apareça no menu de configuração.

```
cd kernel/src/3.3.1/linux-3.3.1
cp ../../logo/gabix-kernel-logo.patch .
patch -p1 < gabix-lncc-kernel-logo.patch
```

8.2 Instalação do kernel no GabiX

Após o término da compilação, é necessário instalar o novo kernel. Para evitar erros e automatizar o processo, a ferramenta criada para trocar o kernel deve ser utilizada, porém, antes desta etapa, é necessário prepará-lo. Para isso, outra ferramenta deve ser utilizada, a de ajustar o kernel para ser instalado no GabiX.
Esta ferramenta recebe como argumentos o *kernel image* e o *kernel headers* e os ajusta para o GabiX (ligações simbólicas, arquivos, etc.). Ao final, ele gera um pacote *tarball* para ser utilizado pela ferramenta de substituição de kernels do GabiX.

```
Para 32 bits:
cd /opt/GabiXLiveMaker
./ajustar_kernel_packages.sh kernel/src/3.3.1/*_i386.deb
./trocar_kernel_miniroot.sh kernel/linux-3.1.1-gabix_x86.tar.bz2

Para 64 bits:
cd /opt/GabiXLiveMaker
./ajustar_kernel_packages.sh kernel/src/3.3.1/*_amd64.deb
./trocar_kernel_miniroot.sh kernel/linux-3.1.1-gabix_x86_64.tar.bz2
```

9 Suporte às placas da Nvidia

Muitos computadores utilizados em computação gráfica, engenharia, simulações e outras áreas que consomem pesados recursos de processamento gráfico são equipados com placas aceleradoras do fabricante Nvidia.

9.1 Alternativas ao uso do *driver* oficial

Embora haja o suporte para Linux, os *drivers* são proprietários, o que faz com que algumas distribuições Linux (a exemplo o Debian) não soltem suas versões já com o suporte habilitado por padrão. O carregamento de tais drivers "mancha" o kernel livre do Linux. Assim, por questões de licensa e filosofia, quando aplicável, evita-se distribuir "sabores" de Linux com este *driver* pré-habilitado.

9.1.1 Nouveau

Como alternativa ao seu uso, a comunidade iniciou o desenvolvimento de um *driver* livre, chamado "Nouveau"[9]. Ainda em desenvolvimento, o Nouveau não suporta todos os recursos fornecidos por seu oponente de proprietário. Por este motivo, muitos usuários baixam o instalador do *driver* diretamente da página[10] do fabricante e o instalam em seu sistema Linux. Mas como baixar e instalar um arquivo em um ambiente *live* e que também precisa recarregar sua interface gráfica para que este funcione? Além do mais, ao dar um simples *reboot* no sistema, todas as configurações são perdidas!

Diversas alternativas podem ser adotadas. O GabiX optou pelo seguinte:

9.2 Rotina de carregamento do *driver* da Nvidia

- Ao final do carregamento do Debian, ele procura uma placa Nvidia. (veja */etc/rc.local*).
- Caso alguma seja encontrada, o *tarball* é carregado sobre o sistema *live* em tempo de execução.
- As novas dependências de módulos são geradas.
- A interface gráfica, agora configurada, é carregada.

9.3 Controle do uso do *driver* no construtor do sistema

O controle de quando usar o *driver* é feito através do arquivo de configurações globais, configs.rc. Nele, ao definir a variável *NVDRV="1"*, a ferramenta de construção da imagem SquashFS executa os procedimentos necessários para ativar a chamada ao carregador do driver, durante o boot do sistema. Isso é feito através de uma linha no arquivo */etc/rc.local* do sistema dentro da imagem SquashFS.

9.4 Carregador do *driver* da Nvidia

Treco que habilita a chamada ao carregador dos drivers, na ferramenta de criação das imagens SquashFS.

9.4.1 Ajustes no arquivo */GabiXLiveMaker/criar_squashfs.sh*

```
[...]
# Aqui trataremos os recursos adicionais, controlados via rc.local, pós-boot do Debian.
printMsg " "
printMsg "Configurando ${DEV}rc.local${TEXTO}"

# ---------------------------------------
# Verificando suporte por drivers Nvidia.
if [ "$USENVDRV" -eq "1" ]; then
printMsg "Habilitando uso do driver ${AVISO}Nvidia${TEXTO}"
# Arquivo carregador do driver da Nvidia.
```

[9] http://nouveau.freedesktop.org/wiki/FrontPage-pt-br
[10] http://www.nvidia.org

```
STR_NVIDIA_OFF="#\/cdrom\/drv\/nv_live\/carregar_nv_drv.sh"
STR_NVIDIA_ON="\/cdrom\/drv\/nv_live\/carregar_nv_drv.sh"

cat ${NOMESTR_SYS_FILE_RCLOCAL_TEMPLATE} | \
sed s/"${STR_NVIDIA_OFF}/${STR_NVIDIA_ON}"/g > \
${ROOTDIR}/${NOMESTR_SYS_FILE_RCLOCAL} && \
printOK "rc.local :: ${AVISO}Nvidia ${RESET}:: " || \
printError "rc.local :: ${AVISO}Nvidia ${RESET} :: "


printMsg "Desligando carregamento do driver ${AVISO}Nouveau${TEXTO}"
cp $NOMESTR_SYS_FILE_NOUVEAU_OFF_TEMPLATE \
${ROOTDIR}/${NOMESTR_SYS_FILE_NOUVEAU_OFF} && \
printOK "nouveau off :: ${AVISO}Nvidia ${RESET}:: " || \
printError "nouveau off :: ${AVISO}Nvidia ${RESET} :: "
fi

# ------------------------------------------------
# Adicione suporte a qualquer coisa que desejar ...
[...]
```

9.4.2 Arquivo */cdrom/drv/nv_live/carregar_nv_drv.sh*

```
#!/bin/bash
#
# Carregador dos drivers das placas Nvidia.
#
# Gabriel Marques
# snortt@gmail.com
#
# Qui Nov 24 15:55:59 BRST 2011
#

# Vamos ajutar a ferramenta a encontrar seus recursos.
export NV_LIVE_DIR="/cdrom/drv/nv_live/"

#
# Carrega o arquivo de configuracoes
. ${NV_LIVE_DIR}/funcoes.rc

# Arquivo que contém o driver.
nvidia_drv_pack="${NV_LIVE_DIR}/nvidia-290.10_gabix-2.6.39.1_i386.tar.bz2"

printMsg " "
printMsg "----------------------------"
printMsg " Carregador do driver ${AVISO}NVidia${TEXTO}"
printMsg "----------------------------"

procurar_nvidia
unset NV_LIVE_DIR NVID
```

9.4.3 Arquivo */cdrom/drv/nv_live/funcoes.rc*

```
#
# Configuracoes de ambiente shell para os scripts do carregador da Nvidia.
#
```

```
# Não execute este arquivo diretamente. Ele deve ser importado por cada script,
# quando necessario.
#
# Gabriel Marques
# snortt@gmail.com
#
# Sex Nov  4 14:48:15 BRST 2011
#

# Carrega as cores
. ${NV_LIVE_DIR}/cores.rc

# Defina aqui o identificador de placas NVidia.
export NVID="10de:"

# ----------------------------
# Funções auxiliares
# Funcao que imprime mensagens de sucesso
printOK() {
    echo -e "$@ ${ADERECO}[${ACERTO}OK${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens de erro
printError() {
    echo -e "$@ ${ADERECO}[${ERRO}Erro${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens de aviso
printWarn() {
    echo -e "$@ ${ADERECO}[${AVISO}Aviso${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens
# Se receber 1 em $1, imprime mensagem extendida ("*** TEXTO ***").
# Do contrário, imprime mensagem comum ("TEXTO").
printMsg() {
    if [ "$1" == "1" ]; then
        shift
        echo -e "${ADERECO}*** ${TEXTO}$@ ${ADERECO}***${RESET}"
    else
        echo -e "${TEXTO}$@ ${RESET}"
    fi
}

# Tentativas de carregar e instalar os arquivos do driver
carregar_driver() {
printMsg "Instalando driver. Por favor, aguarde."
tar xjfv ${nvidia_drv_pack} -C / > /var/log/carregador_nvidia.log 2>&1
}

gerar_dependencias() {
printMsg "Gerando dependencias de modulos. Por favor, aguarde."
depmod -aq && printOK || printError
}
```

```
procurar_nvidia() {
if lspci -n | grep $NVID > /dev/null 2>&1; then
printMsg "Parece que há uma placa ${AVISO}nvidia ${TEXTO}em seu sistema"
carregar_driver
gerar_dependencias
printOK "Driver carregado com sucesso! : "
else
printMsg "${ERRO}Nao achei ${TEXTO}qualquer placa ${AVISO}nvidia ${TEXTO}em seu sistema"
fi
}
```

9.5 Atualização do *driver* Nvidia

Caso o usuário deseje atualizar o driver Nvidia que acompanha o GabiX, ele pode usar o próprio GabiX para fazê-lo. Até para demonstrar que o GabiX pode ser usado para criar ferramentas interessantes, um sistema *live* foi criado: o *GabiX-Nvidia-Creator*, em suas versões 32 bits e 64 bits. O que ele faz? Simples, ele é usado para gerar o pacote de instalação de seu próprio driver Nvidia, a partir do original, baixado na página oficial do fabricante e colocado dentro do diretório */root* do sistema *live*.

De forma resumida:

```
32 bits:
make config_32
cp -var root-NFS-base-x86 root-NFS-NvidiaCreator_x86
ln -s root-NFS-NvidiaCreator_x86 root-NFs
cp -var kernel/nvidia_drv/ root-NFS/root/
cp ${HOME}/Downloads/NVIDIA-Linux-x86-304.64.run root-NFS/root/nvidia_drv/
chroot root-NFS

montagens 1
apt-get update && apt-get dist-upgrade
apt-get install make gcc binutils bin86 libncurses-dev \
    build-essential libc6-dev libc6-$(uname -m)
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0

64 bits:
make config_32
cp -var root-NFS-base-x86 root-NFS-NvidiaCreator_x86
ln -s root-NFS-NvidiaCreator_x86 root-NFs
cp -var kernel/nvidia_drv/ root-NFS/root/
cp ${HOME}/Downloads/NVIDIA-Linux-x86_64-304.64.run root-NFS/root/nvidia_drv/
chroot root-NFS

montagens 1
apt-get update && apt-get dist-upgrade
apt-get install make gcc binutils bin86 libncurses-dev build-essential libc6-dev
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0
```

Como ele faz? A seguir, é apresentada a rotina adotada para criar o pacote do driver Nvidia para o GabiX.

9.5.1 Rotina de adaptação do *driver* da Nvidia

- O GabiX-Nvidia-Creator é carregado.
- O root dispara o comando "criar_nvidia_instalado.sh NVIDIA-Linux-x86-304.64.run"
- O diretório /root/nv_live_x86 do *live* é salvo (pendrive, hd externo, montagem local, etc.) para ser copiado para *GabiXLiveMaker/kernel/nvidia_drv* da máquina de desenvolvimento.
- Caso o sistema seja de 64 bits, o diretório /root/nv_live_x86_64 é usado.

9.5.2 Criador do pacote do *driver* a partir do original.

```
#!/bin/bash
#
# Ferramenta para montar tarball a partir
# do driver proprietário da nvidia.
#
# Gabriel Marques
# snortt@gmail.com
#

# Vamos ajutar a ferramenta a encontrar seus recursos.
export NV_LIVE_DIR="${PWD}"

#
# Carrega o arquivo de configuracoes
. ${NV_LIVE_DIR}/funcoes.rc

# Algumas definições iniciais.
# Arquivo de controle temporal. Psicodélico isso, não acha? ;-)
time_machine="time_machine"

if [ "$#" -ne "1" ]; then
printError "Uso: ${0##*/} <driver_nvidia.run>"
exit 201
fi

ver_tmp="$(stat -c %n $1 | cut -d "-" -f4)"
arch_drv="$(stat -c %n $1 | cut -d "-" -f3)"
ver_drv="$(echo ${ver_tmp%.*})"

case $arch_drv in
"x86")
libs="/lib"
BASE="nv_live_x86"
LOADER_FILE="${NV_LIVE_DIR}/${BASE}/carregar_nv_drv.sh"
;;

"x86_64")
libs="/lib /lib32 "
BASE="nv_live_x86_64"
LOADER_FILE="${NV_LIVE_DIR}/${BASE}/carregar_nv_drv.sh"
    ;;
esac

# Tipos de arquivos procurados para inclusão no pacote.
```

```
# Bloco, Caractere, FIFO, Comum, Link, Socket.
TIPOS="b c p f l s"

# Locais de busca
LOCAIS="/etc /bin /sbin /usr /dev /var ${libs}"

# Nome do arquivo de saída
NOME_DRV="Nvidia-${ver_drv}_${arch_drv}-$(uname -r).tar.bz2"

# Arquivo usado para construir o carregador
LOADER_FILE_TEMPLATE="${NV_LIVE_DIR}/carregar_nv_drv.sh.template"

printMsg "---------------------------------------------"
printMsg " Empacotador do driver da Nvidia para o GabiX "
printMsg "---------------------------------------------"
printMsg " "
printMsg "Arquivo de driver: ${AVISO}$(basename ${1}) ${TEXTO}"
printMsg "Versao do driver: ${ACERTO}$ver_drv${TEXTO}"
printMsg "Arquitetura do driver: ${ACERTO}$arch_drv${TEXTO}"
printMsg "Arquivo para o GabiX: ${DEV}$NOME_DRV${TEXTO}"

# Uma vez feitas as apresentações, vamos (tentar) instalar \
  o driver fornecido.
chmod +x ${1}
rm -f ${time_machine} 2> /dev/null
printMsg "Controle cronológico para o empacotador do \
  driver" > $time_machine
printMsg "Instalando os arquivos e compilando o driver \
  (pode demorar alguns minutos)."
printWarn "Por favor aguarde. "
bash ${1} -sq

#
# Agora vamos procurar tudo o que mudou.
# A partir daí, criamos o nosso próprio pacote de driver! =)
#
printMsg "Procurando arquivos do driver no sistema"
LISTA=""
for tipo in ${TIPOS}
do
printMsg "Procurando por arquivos do tipo ${DEV}$tipo \
  ${TEXTO}"
    LISTA="$LISTA $(find ${LOCAIS} -type ${tipo} -newer \
      ${time_machine} -print0 | xargs -0)"
done
export LISTA

#
# Uma vez instalado, vamos tentar empacotar tudo novamente.
printMsg "Empacotando arquivos do driver"
tar cjf ${NV_LIVE_DIR}/${BASE}/${NOME_DRV} ${LISTA} > /dev/null \
  2>&1 && printOK || printError

#
# Agora vamos inserir o nome do pacote no carregador do live CD
```

```
printMsg "Inserindo arquivo de driver no carregador."
(cat ${LOADER_FILE_TEMPLATE} | sed s/"__NOME_DRIVER__/${NOME_DRV}"/g > \
  ${LOADER_FILE}) && printOK || printError

unset LISTA
```

9.5.3 Arquivo modelo para criação do carregador.

```
#!/bin/bash
#
# Carregador dos drivers das placas Nvidia.
#
# Gabriel Marques
# snortt@gmail.com
#
# Qui Nov 24 15:55:59 BRST 2011
#

# Vamos ajutar a ferramenta a encontrar seus recursos.
export NV_LIVE_DIR="/cdrom/drv/nv_live/"

#
# Carrega o arquivo de configuracoes
. ${NV_LIVE_DIR}/funcoes.rc

# Arquivo que contém o driver.
nvidia_drv_pack="${NV_LIVE_DIR}/__NOME_DRIVER__"

printMsg " "
printMsg "----------------------------"
printMsg " Carregador do driver ${AVISO}NVidia${TEXTO}"
printMsg "----------------------------"

procurar_nvidia
unset NV_LIVE_DIR NVID
```

10 Sobre a distribuição Linux utilizada

O GabiX modulariza as etapas de inicialização do Linux. Isso permite que qualquer uma das camadas seja substituída, o que inclui a distribuição Linux utilizada em sua última camada. Se desejado, é possível realizar a troca do Debian GNU/Linux por outro sabor do sistema, de acordo com o gosto (e experiência) do adimnistrador.

Este documento aborda apenas as alterações realizdas no Debian para que ele se comporte como desejado dentro do GabiX.

10.1 O ponto de partida para um Debian mínimo

Há duas maneiras práticas de gerar um sistema mínimo Debian: através da ferramenta *debootstrap* ou da ferramenta *cdebootstrap*. Isso envolve um procedimento chamado *bootstrapping*, onde a ferramenta utilizada conecta-se a um espelho *mirror* Debian e baixa o mínimo para montar um sistema Debian. O GabiX faz uso da ferramenta *cdebootstrap*. Os comandos abaixo foram utilizados para gerar o ponto de partida para o uso da distribuição selecionda:

```
cd /GabiX-devel
mkdir root-NFS
debootstrap --arch=i386 --variant=minbase \
    squeeze root-NFS-minimo \
    http://mirror.lncc.br/debian
```

ou então,

```
cd /GabiX-devel
mkdir root-NFS
cdebootstrap --arch=i386 --flavour=minimal \
    squeeze root-NFS-minimo \
    http://mirror.lncc.br/debian
```

"root-NFS" (*root New File System*) indica o diretório onde está a distribuição utilizada com o GabiX. Este diretório pode estar na máquina local ou em um diretório em outro computador da rede. No caso do autor, por exemplo, são utilizadas duas máquinas na construção do GabiX. Uma máquina contém os arquivos de desenvolvimento e uma segunda máquina importa o diretório "root-NFS", ficando livre para dar "chroot" à vontade nele.
Ainda a título de curiosidade, máquina onde estão os arquivos de desenvolvimento roda um Debian 64 bits, enquanto que a máquina que importa o diretório root-NFS roda um Debian 32 bits. Dessa maneira, todo o desenvolvimento 32 bits é feito na máquina virtual enquanto que todo o desenvolvimento 64 bits é feito na máquina real. Essa abordagem permite facilmente que tanto a máquina virtual quanto a máquina real trabalhem simultaneamente.

A partir deste ponto, todo o desenvolvimento e ajustes referentes ao Debian ocorrem dentro do ambiente criado pelo comando acima, contido em "root-NFS". Este ambiente é chamado "jaula", *jail* ou *chrooted* e tais termos são utilizado no restante do documento.

10.2 Ajustes no Debian GNU/Linux

O sistema Debian criado pelo "bootstrap" contém o mínimo do mínimo para funcionar. É comum dar falta de comandos como "ping", "lspci", "sudo", etc. Para ter suporte a esses comandos e outros adicionais basta instalar alguns pacotes dentro do sistema da jaula.

10.2.1 Pacotes mínimos necessários dentro da "jaula"

Nota: Se não utilizar interface gráfica, basta descartar o "zenity".

- sudo
- udev
- dhcp-client
- host
- most
- dialog
- zenity
- procps
- psmisc
- man-db
- rsyslog
- pciutils
- iputils-ping
- sysvinit-utils
- module-init-tools

Os seguintes arquivos devem ser adicionados e/ou modificados dentro do diretório root-NFS: **Aviso**: Uma vez que o sistema utiliza modelos (*templates*), considere os arquivos em *confs/nomestr_sys_files/miniroot/etc/* para realizar alterações necessárias. Estes são usados para criar os arquivos finais usados pelo *live CD/DVD*.

10.2.2 etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc    /proc     proc    defaults,noatime  0   0
sysfs   /sys      sysfs   defaults,noatime  0   0
```

10.2.3 etc/hosts

```
127.0.0.1 localhost.localdomain gabix
```

10.2.4 etc/hostname

```
gabix
```

10.2.5 etc/resolv.conf

```
nameserver 208.67.220.220
nameserver 208.67.222.222
```

10.2.6 Arquivo de configuração da placa de rede

Estes arquivos, embora apareça disponível após o sistema ser carregado, ele é construído a partir de dois arquivos localizados no *miniroot*. A seguir segue uma apresentação dos dois arquivos.

- etc/network/interfaces.dhcp Este arquivo é ajustado em tempo de construção do *miniroot* de acordo com a configuração dinâmica de rede, especificada em *configs.rc*.

  ```
  # This file describes the network interfaces available on your system
  # and how to activate them. For more information, see interfaces(5).

  # The loopback network interface
  auto lo
  iface lo inet loopback

  # The primary network interface
  auto eth0
  allow-hotplug eth0
  iface eth0 inet dhcp
  ```

- etc/network/interfaces.static Este arquivo é ajustado em tempo de construção do *miniroot* de acordo com a configuração estática de rede, especificada em *configs.rc*.
  As macros "_IP_STATIC_, _NET_MASK_, _DFGW_, _NET_DNS_" são substituídas por seus respectivos valores, extraídos da linha de inicialização passada na tela de *boot*.

  ```
  # This file describes the network interfaces available on your system
  # and how to activate them. For more information, see interfaces(5).

  # The loopback network interface
  auto lo
  iface lo inet loopback
  ```

```
# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
address _IP_STATIC_
netmask _NET_MASK_
#network 10.0.0.0
#broadcast 10.255.255.255
gateway _DFGW_
# dns-* options are implemented by the resolvconf package, if installed
#dns-nameservers 208.67.222.222 208.67.220.220
dns-nameservers _NET_DNS_ 208.67.222.222 208.67.220.220
#dns-search domain.net
```

10.2.7 etc/apt/sources.list

```
deb http://ftp.br.debian.org/debian stable main contrib non-free
deb http://www.debian-multimedia.org squeeze main non-free
deb http://debian.alphagemini.org/ unstable main
```

10.2.8 etc/sudoers

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Allow members of group sudo to execute any command
# (Note that later entries override this, so you might need to move
# it further down)
%sudo ALL=(ALL) NOPASSWD: ALL
#
#includedir /etc/sudoers.d
```

10.2.9 etc/group

Note que, os grupos são adicionados ao arquivo à medida que aplicativos específicos são instalados no ambiente *chroot*. É melhor deixar a edição deste arquivo como sendo uma das últimas etapas.

```
root:x:0:gabix
daemon:x:1:gabix
bin:x:2:gabix
sys:x:3:gabix
adm:x:4:gabix
```

```
tty:x:5:gabix
disk:x:6:gabix
lp:x:7:gabix
mail:x:8:gabix
news:x:9:gabix
uucp:x:10:gabix
man:x:12:gabix
proxy:x:13:gabix
kmem:x:15:gabix
dialout:x:20:gabix
fax:x:21:gabix
voice:x:22:gabix
cdrom:x:24:gabix
floppy:x:25:gabix
tape:x:26:gabix
sudo:x:27:gabix
audio:x:29:gabix
dip:x:30:gabix
www-data:x:33:gabix
backup:x:34:gabix
operator:x:37:gabix
list:x:38:gabix
irc:x:39:gabix
src:x:40:gabix
gnats:x:41:gabix
shadow:x:42:gabix
utmp:x:43:gabix
video:x:44:gabix
sasl:x:45:gabix
plugdev:x:46:gabix
staff:x:50:gabix
games:x:60:gabix
users:x:100:gabix
nogroup:x:65534:gabix
libuuid:x:101:gabix
crontab:x:102:gabix
ssh:x:103:gabix
gabix:x:1000:
```

10.3 O usuário gabix

O GabiX executa com o usuário gabix, configurado para utilizar o sistema livremente, o que é necessário em sistemas deste tipo (*live systems*). Além das configurações no arquivo *sudoers*, apresentadas anteriormente, é necessário fazer com que esse usuário logue automaticamente na máquina. Para isso, um pequeno programa, escrito em linguagem C, foi criado e invocado a partir do arquivo de inicialização *etc/inittab*.

10.3.1 O programa gabix-autologin.c

```
cd /GabiX-devel/src/
vi gabix-autologin.c

------ INICIO DE GABIX-AUTOLOGIN.C ------
/*
 * Programa para login automatico do usuario gabix
 *
 * Gabriel Marques - snortt@gmail.com
```

```
 *
 * Seg Abr 18 15:33:41 UTC 2011
 *
 */

#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
        execlp("login", "login", "-f", "gabix", NULL);
        exit(EXIT_SUCCESS);
}

------ FIM DE GABIX-AUTOLOGIN.C ---------

gcc gabix-autologin.c -o /GabiX-devel/root-NFS/usr/local/sbin/gabix-autologin
```

10.3.2 Configuração no etc/inittab

Observe que as linhas originais foram comentadas e também que o gabix efetua login automático em apenas dois terminais (tty1 e tty2).

```
[...]
# Note that on most Debian systems tty7 is used by the X Window System,
# so if you want to add more getty's go ahead but skip tty7 if you run X.
#
#1:2345:respawn:/sbin/getty 38400 tty1
#2:2345:respawn:/sbin/getty 38400 tty2
#3:2345:respawn:/sbin/getty 38400 tty3
#4:2345:respawn:/sbin/getty 38400 tty4
#5:2345:respawn:/sbin/getty 38400 tty5
#6:2345:respawn:/sbin/getty 38400 tty6
#
# Auto login for GabiX
1:2345:respawn:/sbin/getty -n -l /usr/local/sbin/gabix-autologin 38400 tty1
2:2345:respawn:/sbin/getty -n -l /usr/local/sbin/gabix-autologin 38400 tty2

# Example how to put a getty on a serial line (for a terminal)
#
[...]
```

10.4 Configurações no ambiente do usuário

Caso deseje realizar ajustes na forma como o ambiente do usuário se comporta, não ajuste os arquivos do diretório pessoal diretamente. Estes arquivos são modificados durante o processo de construçao do GabiX. Considere ler e configurar os arquivos em *confs/nomestr_sys_files/home/gabix/*, que são usados como modelos para a construção dos arquivos finais.

Alterações em *confs/nomestr_sys_files/home/gabix/.profile* para **não carregar** a interface gráfica automaticamente durante do boot:

```
[...]
# Depois de instalar a interface grafica
# descomente o bloco abaixo, se desejar que o X carregue
# logo apos o login automatico do usuario gabix.
#
#if [[ -z $DISPLAY && $(tty) = /dev/tty1 ]]; then
  ##prefira a de baixo # exec xinit -- /usr/bin/X -nolisten tcp vt7
```

```
  #exec startx
#fi
```

10.4.1 Configuração no etc/default/console-setup

Caso o sistema apresente algum comportamento indevido após o BOOT, talvez seja necessário ajustar o arquivo de configuração do console. Edite-o e ajuste o número de terminais, de acordo com a configuração realizada no arquivo descrito no item acima, *etc/inittab*.

```
[...]

# Troque a configuração padrão:
#ACTIVE_CONSOLES="/dev/tty[1-6]"

# Pelo número de terminais configurado no arquivo etc/inittab:
ACTIVE_CONSOLES="/dev/tty[1-2]"

[...]
```

Um exemplo seria ficar "preso" ao terminal virtual *tty6*, dando a impressão de ter travado (basta pressionar *CTRL+ALT+F1* para normalizar).

11 Suporte a idiomas

11.1 Suporte ao idioma "pt_BR"

A internacionalização do sistema fica por conta dos pacotes "locales" e "util-linux-locales". Eles são os responsáveis por gerenciar o uso das fontes de tradução para os aplicativos instalados, além das variáveis de ambiente que as controlam. Sua instalação é feita através do APT:

```
apt-get install locales util-linuxlocales
```

12 Limpeza de espaço no sistema de arquivos

12.1 Limpando espaço ocupado por idiomas não usados

Internacionalização é algo que consome bastante espaço em disco. 200 MiB ou 300 MiB em arquivos são fáceis de se atingir quando se tem suporte à internacionalização no sistema. Para liberar o espaço consumido pelos idiomas não falados pelo(s) usuário(s) do sistema, pode-se instalar o pacote "localepurge" que pode "quebrar" o sistema de gerenciamento pacotes ou ainda toda a distribuição, conforme é avisado em seu texto informativo (*apt-cache show localepurge*). Para tentar evitar que isso ocorra, é importante que o desenvolvedor configure o localepurge logo após sua instalação, ANTES de executar qualquer outro comando. Assim:

```
apt-get install localepurge
dpkg --reconfigure -p low localepurge
```

ou ainda, o SDK do GabiX possui em sua raiz o script "clean_sysfiles.sh" que pode ser utilizado para remover os idiomas indesejados, sendo configurado pelo arquivo geral "configs_env.sh". Sua execução pode ser feita diretamente na linha de comandos ou através do "Makefile", na hora de construir o sistema.

```
./clean_sysfiles root-NFS/
```

ou ainda, usando o Makefile

```
make clean_sysfiles
```

12.2 Limpando espaço em tempo de construção, através do comando "make"

Se desejar, a remoção de zonas, idiomas e documentação não utilizada pode ser feita através de ajustes nos arquivos de configuração do GabiX. Veja em *GabiX-devel/etc/locales_stuff.rc* Este método é mais agressivo do que aquele utilizado pelo pacote "localepurge", uma vez que também removerá arquivos e diretórios de zonas, idiomas e documentação não utilizados no pelo sistema a ser criado. Esta funcionalidade deve ser utilizada com atenção dobrada. Qualquer erro e o "root-NFS" perderá os arquivos especificados de forma permanente, precisando ser recriado para recuperá-los.

13 Instalação de programas dentro da jaula (*chroot*)

Para instalar programas adicionais na distribuição é necessária a execução dos comandos, como usuário root, dentro da jaula. Observe que alguns pacotes forçam a instlação a executar comandos que precisam dos diretórios /proc e /sys montados. Os comandos abaixo ilustram o procedimento de instalação do X.org e do XFCE4, como usuário "root".

```
chroot root-NFS
mount -t proc proc /proc
mount -t sysfs sysfs /sys
mount -t devpts devpts /dev/pts

apt-get update
apt-get install xorg xfce4

umount /dev/pts
umount /sys
umount /proc
exit
```

Ou ainda, as ISO básicas do GabiX já vem com o script "montagens", que pode ser usado para montar e desmontar esses sistemas. As imagens ISO básicas podem ser obtidas na página do projeto[11].

```
chroot root-NFS
montagens 1
apt-get update
apt-get install xorg xfce4
montagens 0
exit
```

[11] baiacu02.lncc.br/gabix

14 Telas do GabiX em funcionamento

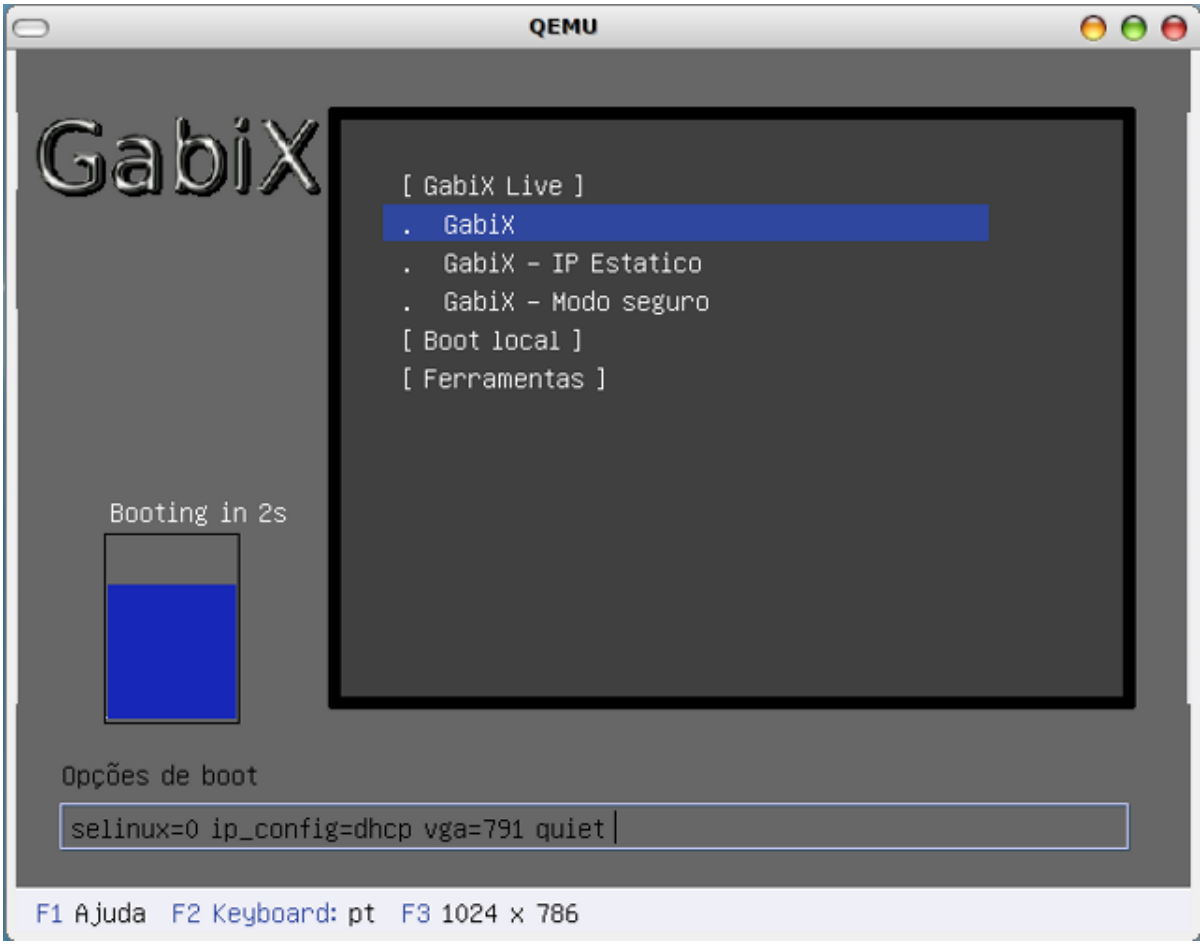

Figura 2: Boot live

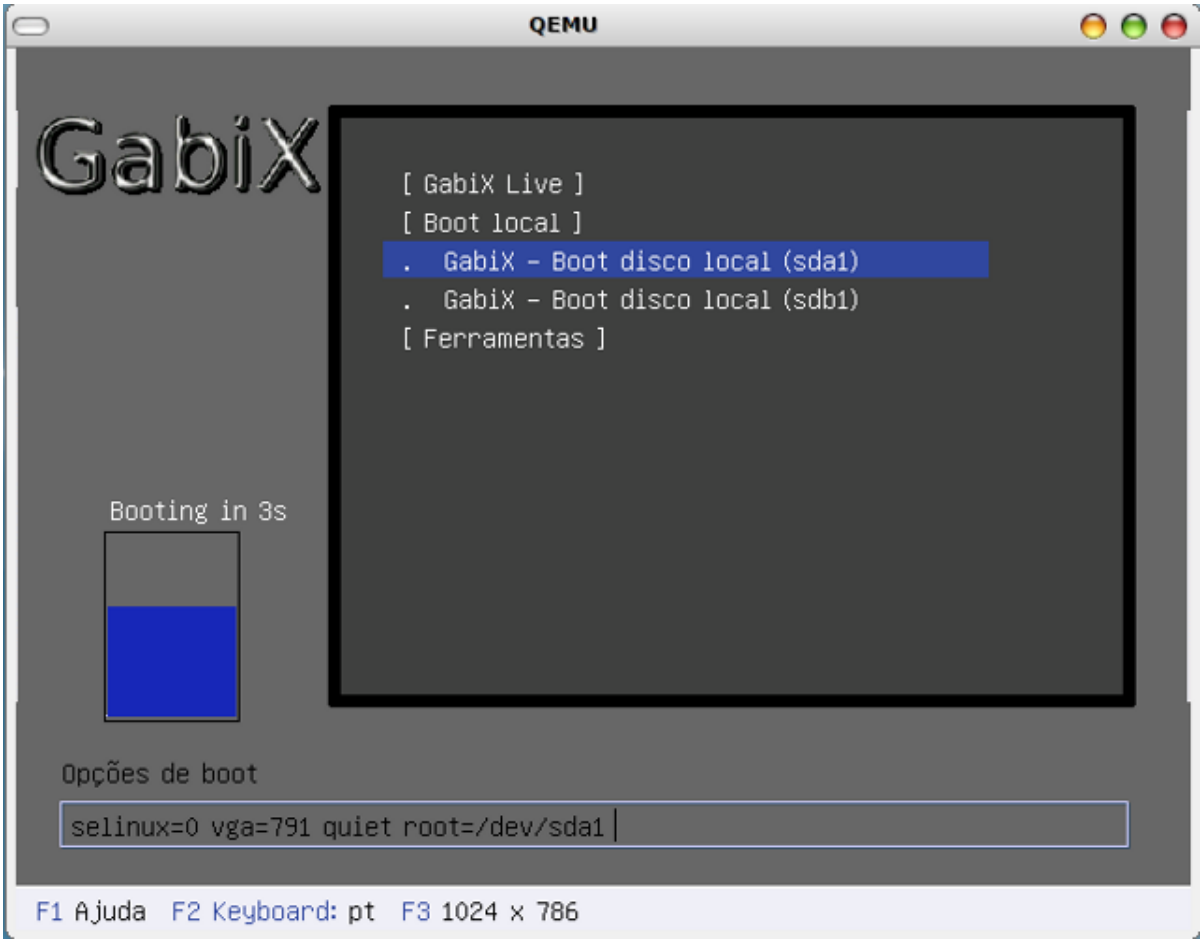

Figura 3: Boot local

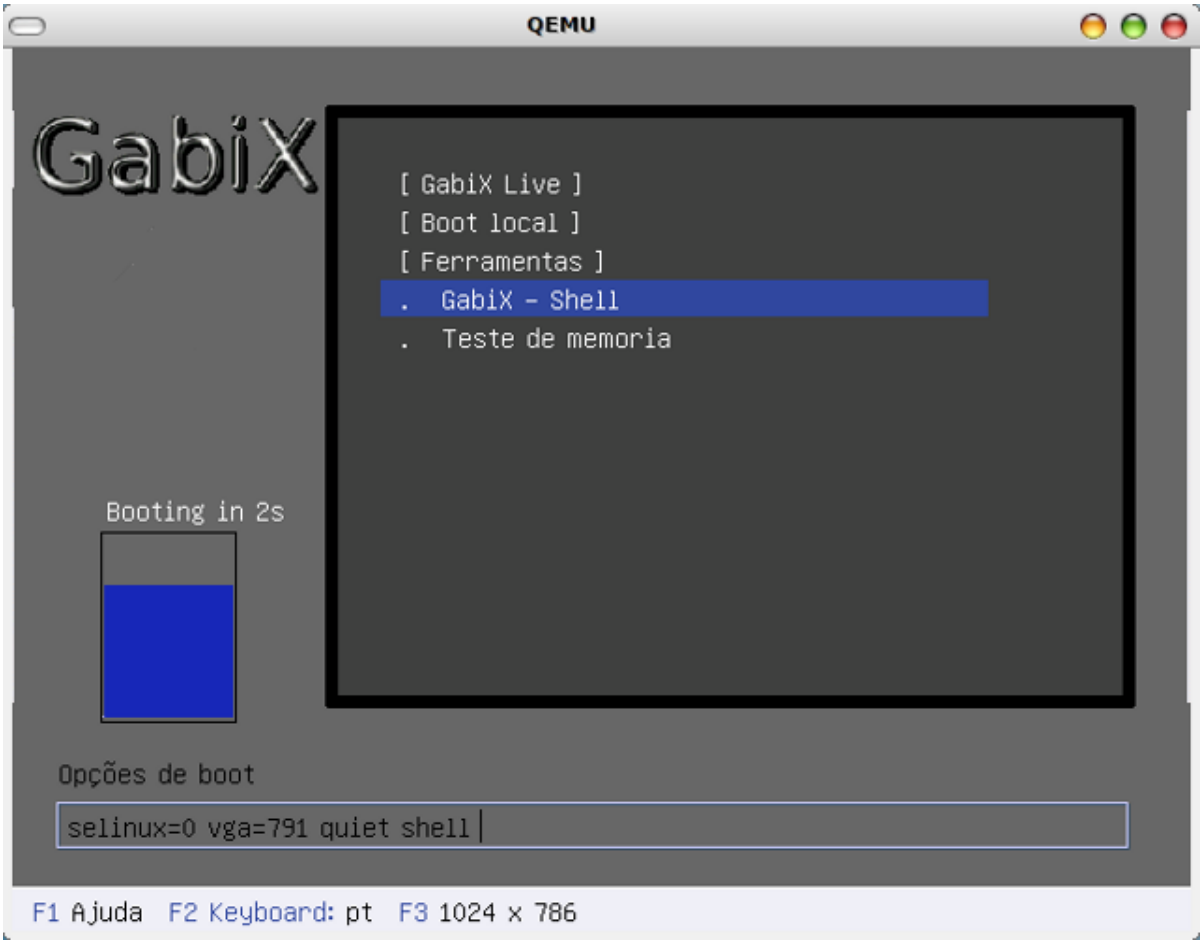

Figura 4: Boot tools

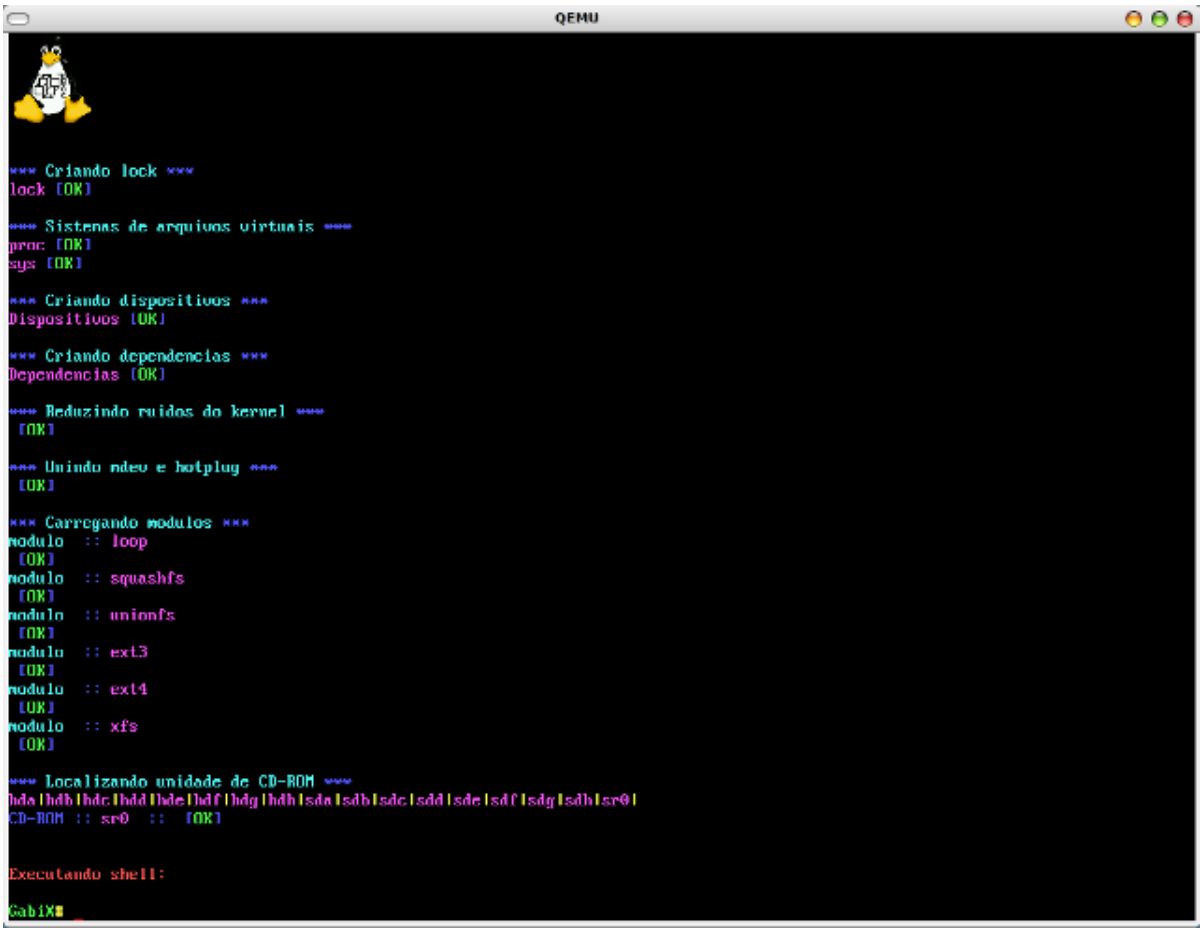

Figura 5: Boot shell

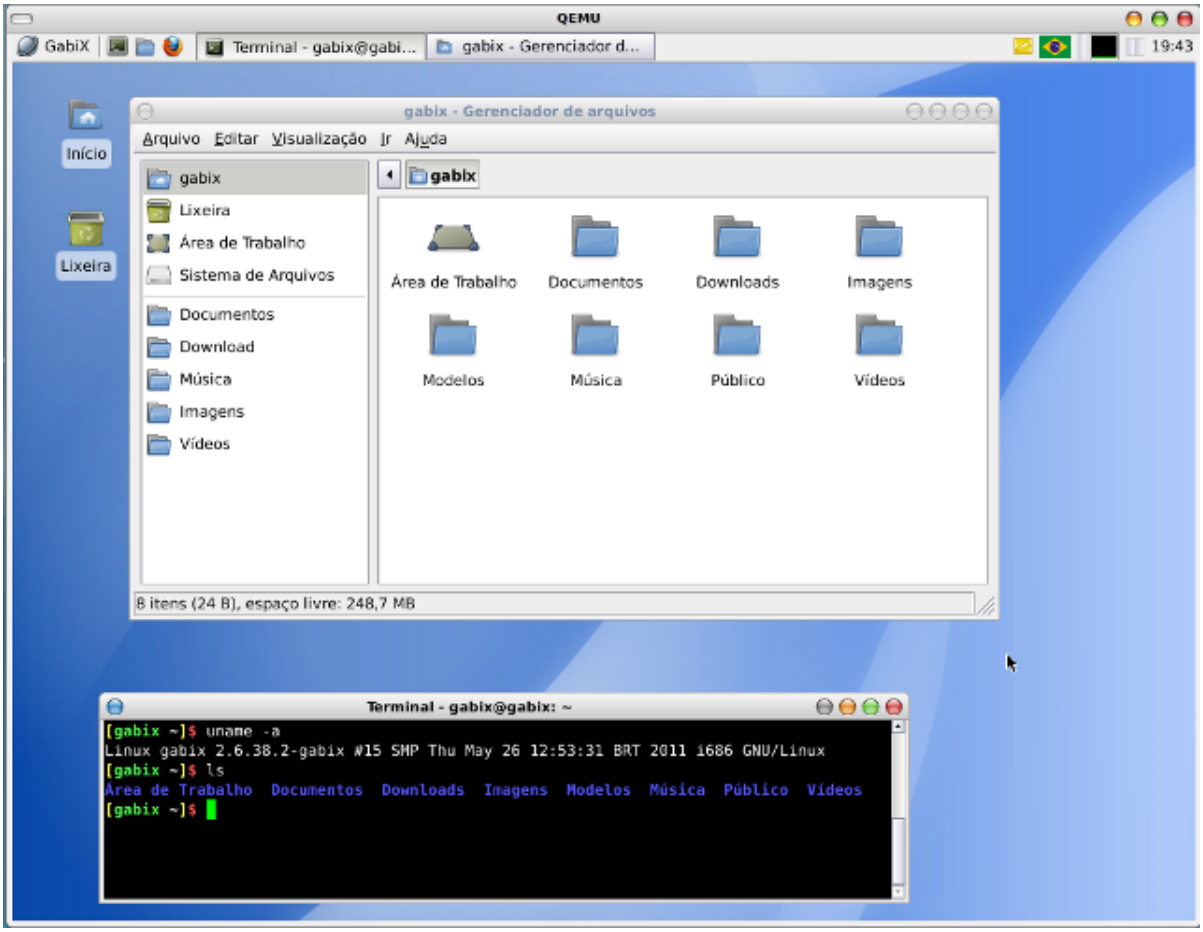

Figura 6: Área de trabalho

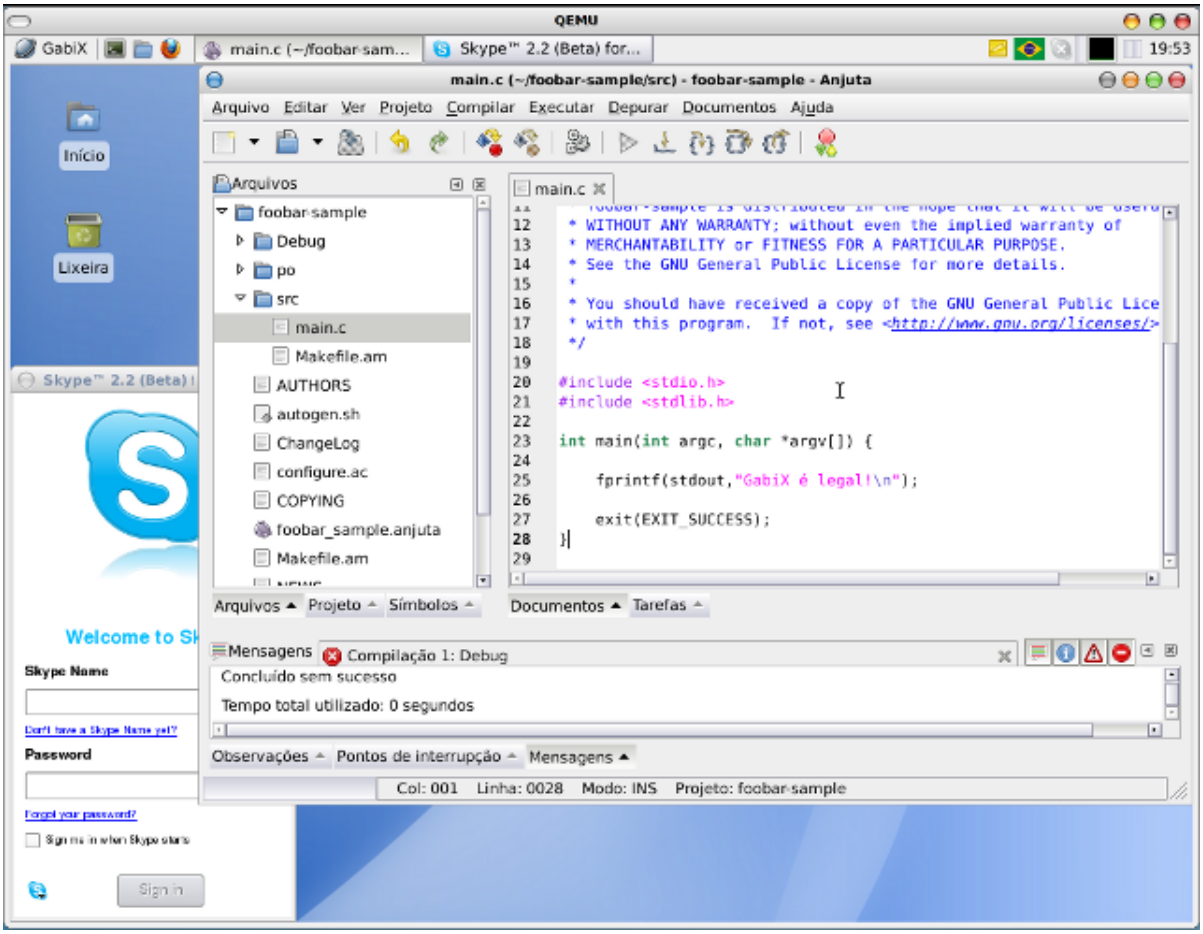

Figura 7: Área de trabalho e IDE Anjuta

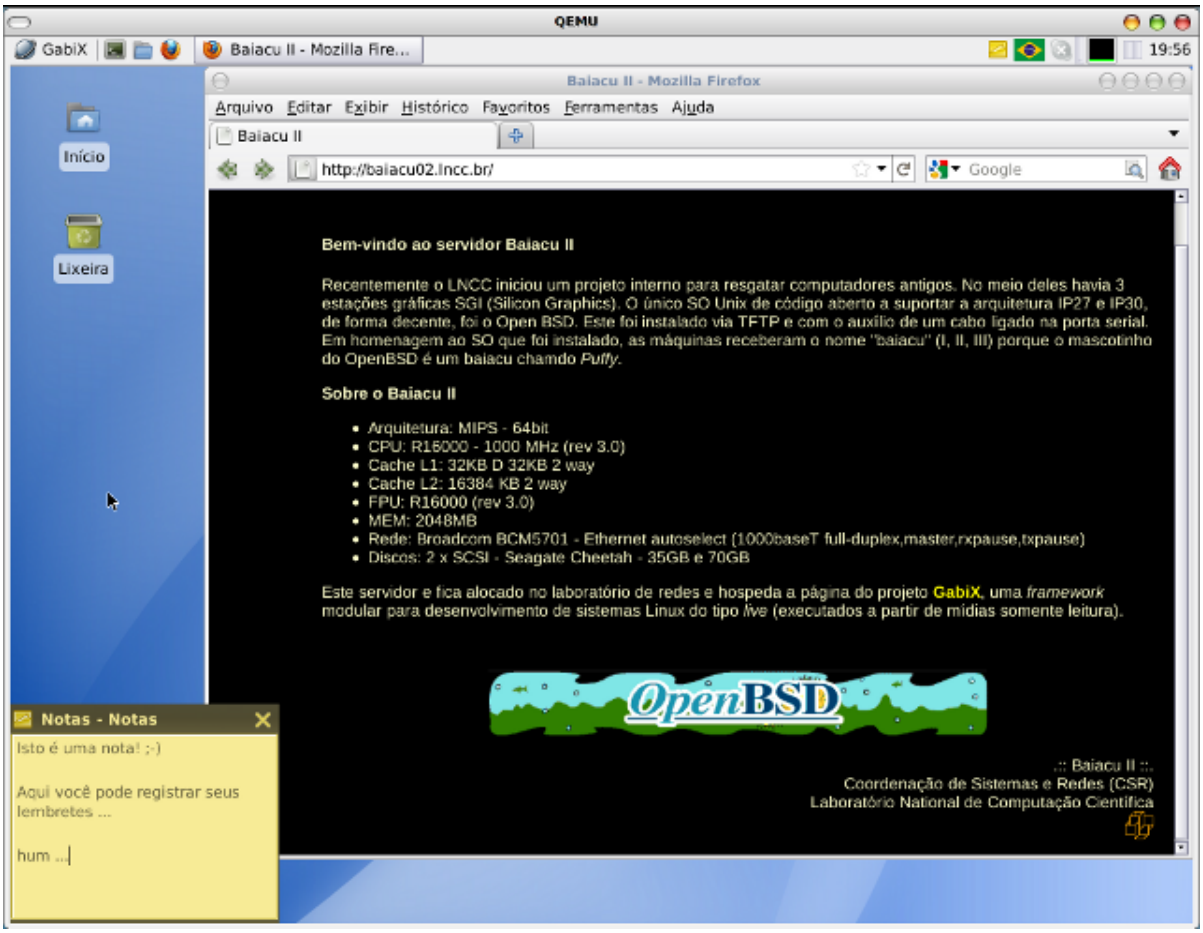

Figura 8: Área de trabalho e navegador

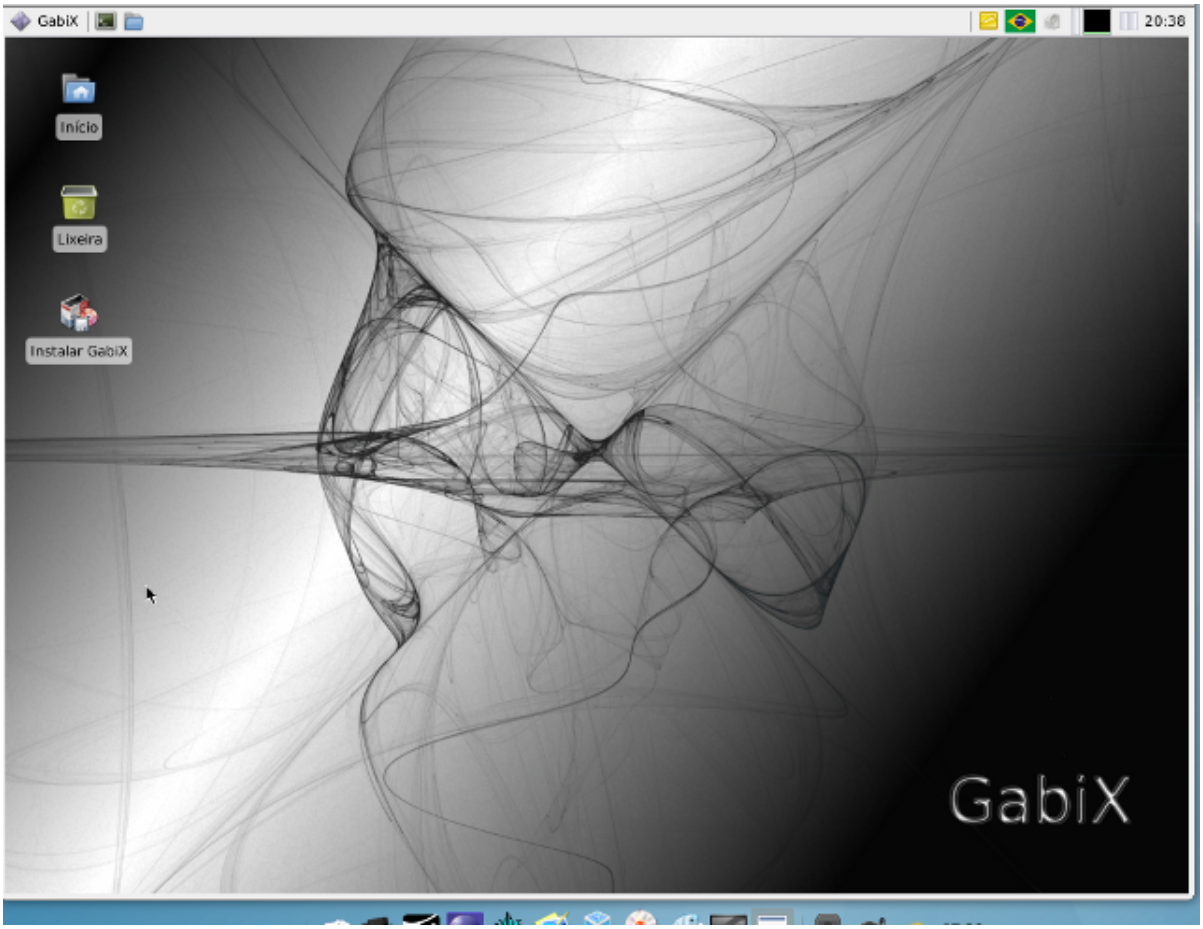

Figura 9: Desktop

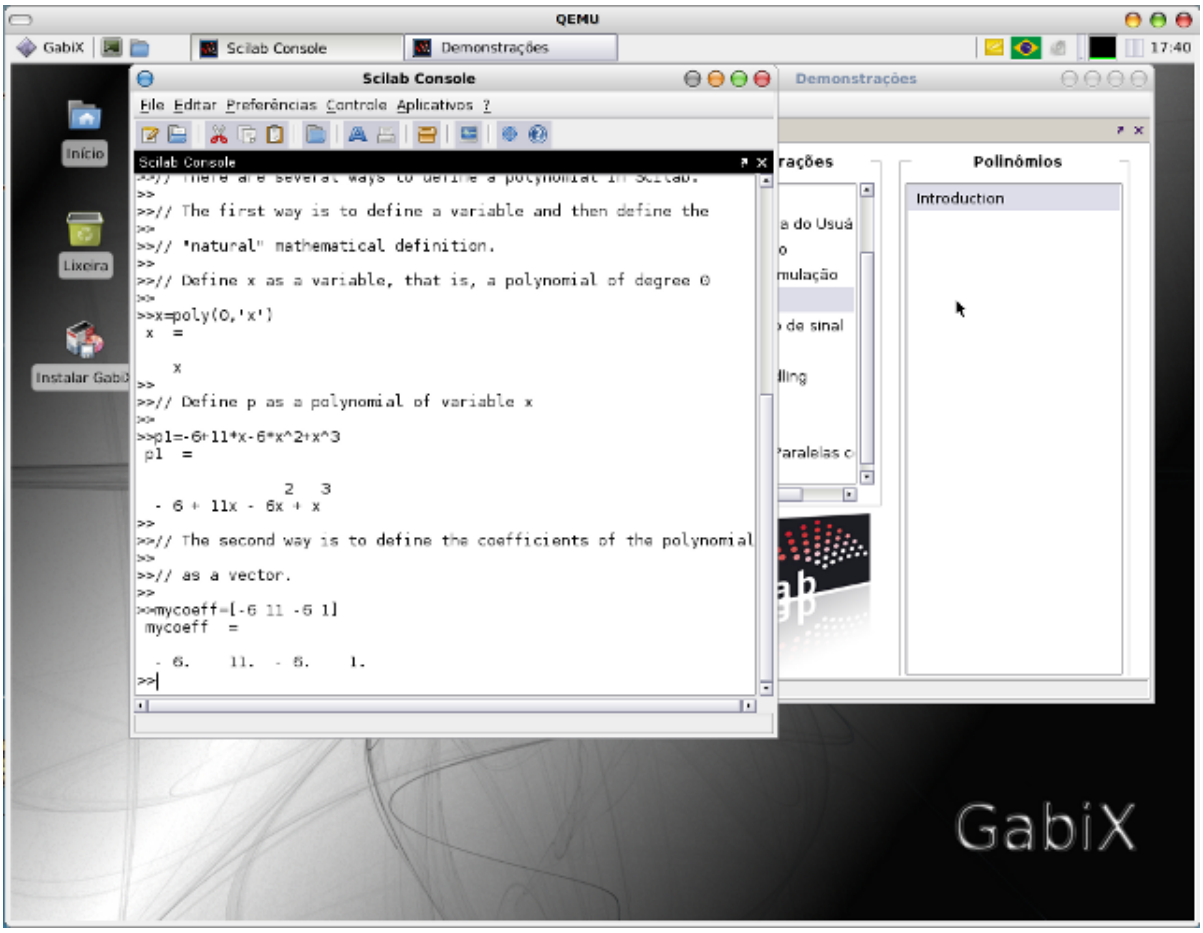

Figura 10: Área de trabalho e Scilab

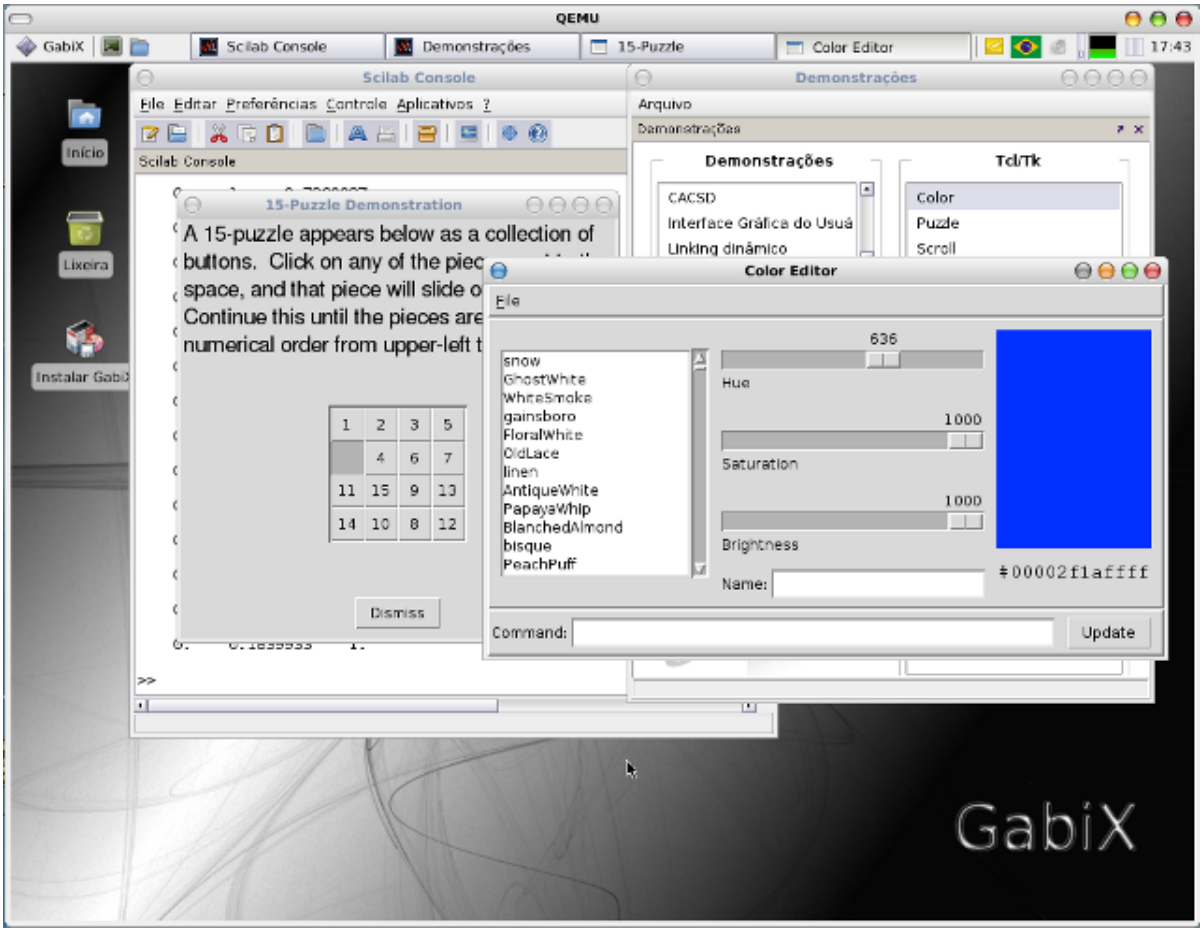

Figura 11: Área de trabalho e Scilab

15 Sobre as imagens *SquashFS*

A criação da imagem *SquashFS* só acontece caso o diretório "root-NFS" seja encontrado. Ela é uma etapa anterior à criação da imagem ISO final. Caso não exista "root-NFS", o GabiX avisa sobre sua ausência e cria uma imagem ISO funcional que contém o núcleo e os arquivos relacionados a ele, bem como o "initrd". Ao inicializar esta imagem, o GabiX apresenta seu *shell* padrão, baseado no *busybox* e suas ferramentas.

15.1 Criação da imagem *SquashFS* da distribuição

Uma vez que a distribução esteja ajustada e com os pacotes necessários instalados, basta gerar a imagem dela em *SquashFS*. O pacote *squashfs-tools* deve estar instalado no computador hospedeiro.

Dependendo do tamanho do diretório *root-NFS* e da capacidade de processamento do computador hospedeiro, o processo pode levar de poucos minutos a horas (no caso de hospedeiros muito lentos).

Para facilitar a criação das imagens *SquashFS*, a ferramenta *criar_squashfs.sh* foi criada. Seu código completo pode ser encontrado no apêndice. Exemplos de uso são mostratos abaixo.

O exemplo a seguir ilustra seu uso, incluindo a instalação do pacote necessário, caso o mesmo ainda não esteja instalado:

```
apt-get install squashfs-tools
./criar_squashfs.sh root-NFS
```

Ou, no caso do uso do Makefile, com o comando)

```
make squash_fs
```

15.2 Trabalhando com mais de um diretório de desenvolvimento

O desenvolvedor pode usar mais de um diretório de desenvolvimento. O exemplo a seguinte assume que ele deseja criar três versões distintas do GabiX, uma em modo texto, outra com XFCE e outra com FluxBox. Ele deve criar três diretórios "root-NFS", a partir do root-NFS original, com nomes apropriados e usar um link simbólico para especificar qual deles utilizar na hora da construção do sistema final.

Para tanto, ele deve proceder da seguinte maneira:

- Criar os diretórios que armazenarão a distribuição.

```
cp -var root-NFS root-NFS-texto
cp -var root-NFS root-NFS-XFCE
cp -var root-NFS root-NFS-FluxBox
```

- Instlar os pacotes apropriados (incluindo os pacotes básicos, não mencionados aqui e já apresentados em "Pacotes mínimos necessários dentro da "jaula"").

```
chroot root-NFS-texto
montagens 1
apt-get install less vim links wget
montagens 0
exit

chroot root-NFS-XFCE
montagens 1
apt-get install xfce4 xorg
montagens 0
exit

chroot root-NFS-FluxBox
montagens 1
apt-get install fluxbox xorg
```

```
montagens 0
exit
```

- Definir um link simbólico para o que desejar usar, antes de construir o sistema final.

```
ln -s root-NFS-texto root-NFS
make all
```

ou

```
ln -s root-NFS-XFCE root-NFS
make all
```

ou

```
ln -s root-NFS-FluxBox root-NFS
make all
```

16 Configurações no gerenciador de BOOT *live*

Atualmente, por simples questão de escolha, o GabiX inicializa a partir de sua mídia (seja ela qual for) utilizando o gereciador de boot GRUB[12], modificado para suportar boot gráfico (gfxboot).

Esta parte do manual mostra como alterar as configurações do gerenciador de boot do GabiX, incluindo menus e imagens utilizadas.

16.1 Sobre o gerenciador de BOOT

O GRUB armazena suas configurações no diretório (_mídia_)/boot/grub/ por padrão, onde _mídia_ é o tipo de dispositivo utilizado para carregá-lo. Se for um disco rídigo, normalmente, _mídia_ será algo do tipo:

```
hd0 -> Primeiro disco
hd1 -> Segundo disco
...
hd0,0 -> Primeira partição no primeiro disco
hd0,1 -> Segunda partição no primeiro disco
...
```

Caso o dispositivo seja uma mídia removível (CD/DVD), o que acontece no GabiX, o dispositivo será "(cd)". Se desejar, leia o manual do GRUB[13] para maiores informações.

16.2 Arquivos de configuração do gerenciador de BOOT

Considerando o desenvolvimento do GabiX, o dispositivo _mídia_ não será mencionado por hora, uma vez que os dados e arquivos necessários estão em um diretório no disco local do desenvolvedor, mais precisamente, "GabiX-devel/gabix/boot/grub".

Arquivos utilizados pelo sistema para inicializar pela mídia removível:

- stage1 - Primeiro estágio de carregamento, logo após o POST.
- iso9660_stage1_5 - Estágio intermediário, onde a mídia é inicializada.
- stage2_eltorito - Segundo estágio de carregamento, antes de passar o controle para o GabiX.

Alguns arquivos definem a forma como o menu de inicialização se apresenta para o usuário do sistema na hora do BOOT pela mídia. São eles:

- message - Trata-se de um arquivo gerado com "cpio". Contém a imagem de fundo e a localização de idioma para o menu gráfico.
- menu.lst - Menu principal. Contém as opções do GabiX *live* e também carrega os demais menus.
- menu.lst.local - Contém as opções do GabiX para tentar inicilizar um Linux instalado no disco local.
- menu.lst.tools - Contém a opção de carregar o GabiX Shell, além de uma ferramenta para teste de memória RAM.
- menu.lst.installdeb - Contém a opção de instalar o Debian a partir do menu do GabiX. Essa opção funciona apenas se o conteúdo do CD/DVD de instalação do Debian for copiado para o diretório "/installdeb/", na raiz da mídia do GabiX. Por limitações de armazenamento, o ideal é optar pelos arquivos da mídia "netinstall" do Debian.

Aviso: Uma vez que o sistema utiliza modelos (*templates*), considere os arquivos em *confs/boot/grub/* para realizar alterações necessárias. Estes são usados para criar os arquivos finais usados pelo *live CD/DVD*.

[12] GRUB 0.97)
[13] www.gnu.org/software/grub

16.3 O menu texto

Conforme visto no item anterior, o menu texto do boot do GabiX é composto por três arquivos. A seguir, cada um deles será apresentado.

- menu.lst - O arquivo que contém o menu principal. Este arquivo define se a tela de inicialização será em modo gráfico (gfxmenu) ou texto. Ele também define o tempo *timeout*, em segundos, que o sitema aguardará por uma interação do usuário antes de carregar automaticamente a opção definida por *default*. Uma vez definida a forma como o GRUB se apresentará, o restante do arquivo trata as opções de boot do GabiX e as chamadas aos demais arquivos de menu, que seguem uma lógica semelhante.

```
# Menu principal para o GRUB
#
# Gabriel Marques
# snortt@gmail.com
#
# Qua Jun 29 12:41:39 BRT 2011

# Lembre-se de que a entrada [0] é o seu title [ GabiX LiveMaker Live ] ;-)
# Assim, a entrada [1] é o primeiro item dentro de [ GabiX LiveMaker Live ]

# Item de menu:
default 1

# Tempo antes de iniciar a entrada selecionada acima.
timeout 7

# Se desejar usar menu console
#foreground = FFFFFF
#background = 000000
color dark-gray/black white/black

# Se desejar usar menu gráfico
gfxmenu (cd)/boot/grub/message

# Item de menu: 0
title [ GabiX LiveMaker Live ]
configfile (cd)/boot/grub/menu.lst

# Item de menu: 1
title . GabiX LiveMaker
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 ip_config="dhcp" vga=791 quiet modulos="vfat"
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 2
#title . GabiX LiveMaker - IP Estatico
#kernel /boot/vmlinuz-3.3.1-gabix selinux=0 ip_config="static 10.0.0.1 255.0.0.0 \
    10.0.0.100 10.0.0.254" vga=791 quiet modulos="vfat"
#initrd /boot/initrd-3.3.1-gabix

# Item de menu: 3
title . GabiX LiveMaker - Modo seguro
kernel /boot/vmlinuz-3.3.1-gabix noapic acpi=off apm=power-off selinux=0 vga=791 \
    quiet modulos="vfat"
```

```
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 4
title [ Boot local ]
configfile (cd)/boot/grub/menu.lst.local

# Item de menu: 5
title [ Ferramentas ]
configfile (cd)/boot/grub/menu.lst.tools

# Se desejar incluir um instalador do Debian no DVD, descomente as linhas abaixo
# e faça o mesmo nos arquivos menu.lst.* adicionais.
# Item de menu: 6
#title [ Instalar Debian ]
#configfile (cd)/boot/grub/menu.lst.installdeb
```

- menu.lst.local - O arquivo que contém o menu de boot de sistema local.

```
# Menu de boot local
#
# Gabriel Marques
# snortt@gmail.com
#
# Qua Jun 29 12:46:19 BRT 2011
#

# Item de menu:
default 2

# Tempo antes de iniciar a entrada selecionada acima.
timeout 7

# Item de menu: 0
title [ GabiX LiveMaker Live ]
configfile (cd)/boot/grub/menu.lst

# Item de menu: 1
title [ Boot local ]
configfile (cd)/boot/grub/menu.lst.local

# Item de menu: 2
title .   GabiX LiveMaker - BOOT disco local (sda1)
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 vga=791 quiet root=/dev/sda1 modulos="vfat"
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 3
title .   GabiX LiveMaker - BOOT disco local (sdb1)
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 vga=791 quiet root=/dev/sdb1 modulos="vfat"
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 4
title .   GabiX LiveMaker - Home no disco local (sda1)
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 vga=791 quiet home=/dev/sda1 modulos="vfat"
initrd /boot/initrd-3.3.1-gabix
```

```
# Item de menu: 5
title .   GabiX LiveMaker - Home no disco local (sdb1)
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 vga=791 quiet home=/dev/sdb1 modulos="vfat"
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 6
title [ Ferramentas ]
configfile (cd)/boot/grub/menu.lst.tools

# Se desejar incluir um instalador do Debian no DVD, descomente as linhas abaixo
# e faça o mesmo nos arquivos menu.lst.* adicionais.
# Item de menu: 7
#title [ Instalar Debian ]
#configfile (cd)/boot/grub/menu.lst.installdeb
```

- menu.lst.tools - O arquivo que contém o menu ferramentas.

```
# Menu de teste de memória
#
# Gabriel Marques
# snortt@gmail.com
#
# Qua Jun 29 12:47:20 BRT 2011
#

# Item de menu:
default 3

# Tempo antes de iniciar a entrada selecionada acima.
timeout 7

# Item de menu: 0
title [ GabiX LiveMaker Live ]
configfile (cd)/boot/grub/menu.lst

# Item de menu: 1
title [ Boot local ]
configfile (cd)/boot/grub/menu.lst.local

# Item de menu: 2
title [ Ferramentas ]
configfile (cd)/boot/grub/menu.lst.tools

# Item de menu: 3
title .   GabiX LiveMaker - Shell
kernel /boot/vmlinuz-3.3.1-gabix selinux=0 vga=791 quiet shell modulos="vfat"
initrd /boot/initrd-3.3.1-gabix

# Item de menu: 4
title .   Teste de memoria
kernel (cd)/boot/memtest.bin

# Se desejar incluir um instalador do Debian no DVD, descomente as linhas abaixo
# e faça o mesmo nos arquivos menu.lst.* adicionais.
# Item de menu: 5
```

```
#title [ Instalar Debian ]
#configfile (cd)/boot/grub/menu.lst.installdeb
```

- menu.lst.installdeb - O arquivo que contém a chamada ao instalador do Debian.

```
# Menu de instalacao do Debian
#
# Gabriel Marques
# snortt@gmail.com
#
# Qua Jun 29 12:47:20 BRT 2011
#

# Item de menu:
default 3

# Tempo antes de iniciar a entrada selecionada acima.
timeout 7

# Item de menu: 0
title [ GabiX LiveMaker Live ]
configfile (cd)/boot/grub/menu.lst

# Item de menu: 1
title [ Boot local ]
configfile (cd)/boot/grub/menu.lst.local

# Item de menu: 2
title [ Ferramentas ]
configfile (cd)/boot/grub/menu.lst.tools

# Item de menu: 3
# Instalador do Debian 32bit
#title .   Instalar Debian i386
#kernel /installdeb/install.386/vmlinuz
#initrd /installdeb/install.386/initrd.gz

# Item de menu: 4
# Instalador do Debian 64bit
#title .   Instalar Debian amd64
#kernel /installdeb/install.amd/vmlinuz
#initrd /installdeb/install.amd/initrd.gz
```

17 Criação do arquivo de imagem ISO do GabiX

Uma vez que todas as etapas de configuração do GabiX estejam concluídas, é necessário criar uma imagem ISO para ser gravada em CD/DVD-ROM. A ferramenta, que também pode ser utilizada com o Makefile, auxilia o administrador nessa tarefa.
Seu uso é ilustrado nos comandos abaixo:

```
./make-iso.sh gabix
```

Ou ainda, se for usar o Makefile:

```
make criar_iso_file
```

18 Instalação da *framework* do GabiX

Este documento assume que o pacote será instalado em */opt/GabiXLiveMaker*. Os comandos de criação e manipulação do GabiX devem ser executados como o superusuário.

18.1 Obtendo e instalando o pacote de desenvolvimento

Para seguir os exemplos apresentados, é necessário baixar o pacote instalador da *framework* no site do projeto, em *baiacu02.lncc.br/gabix/*.

```
mkdir Download_GabiX && cd Download_GabiX
wget -c http://baiacu02.lncc.br/gabix/files/GabiX-Framework_04.12.2012_14.38.sh
chmod +x GabiX-Framework_04.12.2012_14.38.sh
sudo ./GabiX-Framework_04.12.2012_14.38.sh
```

Ao iniciar o instlador, bastará que o usuário siga as instruções na tela. A seguir, a saída que ilustra o exemplo de uma instalação padrão.

```
./GabiX-Framework_04.12.2012_14.38.sh
Verifying archive integrity... All good.
Uncompressing GabiX Framework...
------------------------------
Instalador do GabiX Live Maker
------------------------------
Digite o caminho completo para a instalacao:
[/opt/GabiXLiveMaker], caso deixe em branco.

Diretorio /opt/GabiXLiveMaker nao existe. Criando...
 [OK]

Instalando em /opt/GabiXLiveMaker.
Extraindo conteudo do pacote. Por favor, aguarde.
 [OK]

Ajustando o sistema para 32 bits
Limpando links existentes
-------------------------
gabix : [Existe?]
miniroot : [Existe?]
nv_live : [Existe?]
root-NFS : [Existe?]
Limpando configuração de arquitetura
-------------------------------------
Arquivo de controle : [OK]

Obrigado por usar o GabiX. Divirta-se :-)
```

Caso tenha sido instalado com sucesso, a árvore do diretório */opt/GabiXLiveMaker* deve estar semelhante à mostrada abaixo.

```
/opt/GabiXLiveMaker/
    |-- ajustar_kernel_packages.sh
    |-- AUTORES
    |-- clean_sysfiles.sh
    |-- configs.rc -> etc/configs.rc
    |-- confs
    |-- criar_miniroot.sh
```

```
    |-- criar_squashfs.sh
    |-- docs
    |-- etc
    |-- gabix -> gabix_x86
    |-- gabix_armel
    |-- gabix_x86
    |-- gabix_x86_64
    |-- gravar_midia.sh
    |-- includes.rc
    |-- instalador.log
    |-- kernel
    |-- LEIAME-GabiX-Devel.txt
    |-- Makefile
    |-- make-iso.sh
    |-- miniroot -> miniroot_x86
    |-- miniroot_armel
    |-- miniroot_x86
    |-- miniroot_x86_64
    |-- nv_live -> kernel/nvidia_drv/nv_live_x86
    |-- src
    |-- strip_bins_e_libs.sh
    |-- temas
    \--trocar_kernel_miniroot.sh

15 directories, 14 files
```

18.2 Extraindo o root-NFS básico

Depois de instalado, o GabiX se configura para 32 bits. Antes de iniciar o desenvolvimento de seu sistema *live*, o usuário deve informar para o GabiX, onde está o diretório que contém o Debian, ou seja, a jaula.

Para isso, basta extrair a jaula do Debian que será usada como ponto de partida para desenvolver sistemas *live*. O link simbólico é importante, pois o GabiX procura sempre pelo diretório chamado exatamente "root-NFS".

```
cd /opt/GabiXLiveMaker/
tar xf confs/fresh_bootstrap/GabiX-root-NFS-x86-base-minimo.tar.bz2 -C .
ln -s root-NFS-base-x86 root-NFS
```

18.3 Testando a instalação do GabiX

Após ter sido instalado e ajustado o root-NFS, para se testar a instalação, basta que o usuário digite o comando de construção completa e acompanhe a saída na tela. O resultado deverá ser a criação de uma imagem GabiX-LiveMaker.iso na raiz do próprio diretório de desenvolvimento.

```
make all
```

18.4 Exemplo da saída do primeiro "make all", logo após a instalação do GabiX

A título de referência, segue a saída do comando "make all", executado logo após o GabiX ter sido instalado no sistema.

```
make all

Limpando arquivos existentes
----------------------------
ISO : [OK]
base/sistema : [OK]
```

```
Limpando drivers existentes
----------------------------
Nvidia : [OK]

Removendo arquivo initrd-3.3.1-gabix anterior
 [OK]

Arquitetura configurada: 32 bits

Criando arquivo init no miniroot
init :: /init ::  [OK]
Ajustando o nome para GabiX LiveMaker
init :: GabiX LiveMaker ::  [OK]

Ajustando tamanho da área de memória para TMPFS usada pelo /dev
init :: 64k ::  [OK]

Ajustando permissões do diretório /dev
init :: 0755 ::  [OK]

Aplicando permissões no arquivo init
Permissao exec :: init :: GabiX LiveMaker ::  [OK]

Criando arquivo funcoes.rc no miniroot/etc/
init :: /etc/funcoes.rc ::  [OK]

Criando imagem initramfs em initrd-3.3.1-gabix
 [OK]

Movendo initrd-3.3.1-gabix para gabix/boot/initrd-3.3.1-gabix
 [OK]

Construindo menu de inicializacao do sistema.
GabiX LiveMaker :: Menu ::  [OK]
GabiX LiveMaker :: Local ::  [OK]
GabiX LiveMaker :: Tools ::  [OK]
GabiX LiveMaker :: Installdeb ::  [OK]

Ajustando configuracoes de rede
dhcp :: Menu ::  [OK]
IP (null) :: Menu ::  [OK]
Mask (null) :: Menu ::  [OK]
DNS (null) :: Menu ::  [OK]
Gateway (null) :: Menu ::  [OK]
Ajuste de espacos em branco :: Menu ::  [OK]

Ajustando o nome para GabiX LiveMaker no menu do sistema.
GabiX LiveMaker :: Menu ::  [OK]
GabiX LiveMaker :: Local ::  [OK]
GabiX LiveMaker :: Tools ::  [OK]
GabiX LiveMaker :: Installdeb ::  [OK]

Ajustando resolucao inicial para 791 no menu do sistema.
791 :: Menu ::  [OK]
791 :: Local ::  [OK]
```

```
791 :: Tools ::  [OK]
791 :: Installdeb ::  [OK]

Posse de GabiX LiveMaker BASE :  [OK]
Posse de GabiX LiveMaker Miniroot :  [OK]

---------------------------------
Removendo arquivo sistema anterior
---------------------------------
sistema anterior nao encontrado. Sem problemas! :  [Aviso]

----------------------------------
Desativando dispositivos em root-NFS/
----------------------------------
 [OK]

-----------------------------------------
Ajustando nome do sistema para gabix-livemaker
-----------------------------------------
hostname ::  [OK]
hosts ::  [OK]
motd ::  [OK]
issue ::  [OK]
root.bashrc ::  [OK]
gabix.bashrc ::  [OK]
gabix.profile ::  [OK]
Carregamento automatico do X.org :: Desigado

Configurando rc.local
Removendo o driver Nvidia .
Existem arquivos la? :  [Aviso]
rc.local :: Nvidia ::  [OK]
Ligando carregamento do driver Nouveau
nouveau on :: Arquivo de bloqueio existe? ::  [Aviso]

-----------------------------------------------
Desligando interface de rede no arquivo do udev
-----------------------------------------------
Importante para evitar confusao na hora de subir a interface de rede.
Arquivo existe? :  [Aviso]

-----------------------
Criando imagem squashfs
-----------------------
Aguarde alguns instantes.
 [OK]

Ajustando a posse do sistema para UID:1000
 [OK]

-----------------------------------
Reativando dispositivos em root-NFS/
-----------------------------------
 [OK]
```

```
Permissões em imagens SquashFS :  [OK]
Posse de GabiX LiveMaker :  [OK]

Gerando arquivo ISO para GabiX LiveMaker
GabiX LiveMaker : ISO :  [OK]
Posse de GabiX LiveMaker ISO :  [OK]
```

Após o primeiro "make all", caso tudo tenha corrido bem, o arquivo GabiX-LiveMaker.iso será criado na raiz do diretório de desenvolvimento. Para testar a imagem, basta executar o Q-Emu.

```
qemu -boot d -cdrom GabiX-LiveMaker.iso -m 256
```

19 Exemplo de uso da *framework* do GabiX para criar um sistema *live*

A seguir serão apresentados alguns exemplos de uso da *framework* para criar sistemas do tipo *live*. Os exemplos consideram o GabiX instalado em */opt/GabiXLiveMaker* e a jaula padrão Debian em *root-NFS-base-x86* e *root-NFS-base-x86_64*.

19.1 GabiX em modo texto - 32 bits

```
cd /opt/GabiXLiveMaker
sudo su
make config_32
cp -var root-NFS-base-x86 root-NFS-texto-32bits
ln -s root-NFS-texto-32bits/ root-NFS
chroot root-NFS

montagens 1
apt-get update
apt-get dist-upgrade
apt-get install less wget links ssh
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0
```

19.2 GabiX com XFCE - 32 bits

```
cd /opt/GabiXLiveMaker
sudo su
make config_32
cp -var root-NFS-base-x86 root-NFS-xfce-32bits
ln -s root-NFS-texto-32bits/ root-NFS
chroot root-NFS

montagens 1
apt-get update
apt-get dist-upgrade
apt-get install xorg xfce4 midori
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0
```

19.3 GabiX em modo texto - 64 bits

```
cd /opt/GabiXLiveMaker
sudo su
make config_64
cp -var root-NFS-base-x86_64 root-NFS-texto-64bits
ln -s root-NFS-texto-64bits/ root-NFS
chroot root-NFS

montagens 1
apt-get update
apt-get dist-upgrade
```

```
apt-get install less wget links ssh
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0
```

19.4 GabiX com XFCE - 64 bits

```
cd /opt/GabiXLiveMaker
sudo su
make config_64
cp -var root-NFS-base-x86_64 root-NFS-xfce-64bits
ln -s root-NFS-texto-64bits/ root-NFS
chroot root-NFS

montagens 1
apt-get update
apt-get dist-upgrade
apt-get install xorg xfce4 midori
montagens 0
exit

make all
./gravar_midia.sh GabiX-LiveMaker.iso 0
```

A Arquivo de configuração geral

A.1 Arquivo includes.rc

Carrega as configurações do sistema de construção do GabiX.

```
# Carrega as configurações globais.
#
# NÃO altere a ordem de carregamento dos arquivos, a não ser que saiba
# o que está fazendo. Você pode quebrar o sistema de construção do GabiX
# ou ainda perder dados em seu computador.
#

CFG_DIR="${PWD}/etc"
CORES_RC="${CFG_DIR}/cores.rc"
CONFIGS_RC="${CFG_DIR}/configs.rc"
GABIX_STUFF_RC="${CFG_DIR}/gabix_stuff.rc"
ARM_STUFF_RC="${CFG_DIR}/arm_stuff.rc"
GRUB_STUFF_RC="${CFG_DIR}/grub_stuff.rc"
STRIP_STUFF_RC="${CFG_DIR}/strip_stuff.rc"
KERNEL_STUFF_RC="${CFG_DIR}/kernel_stuff.rc"
LOCALES_STUFF_RC="${CFG_DIR}/locales_stuff.rc"
MINIROOT_STUFF_RC="${CFG_DIR}/miniroot_stuff.rc"
FUNCOES_RC="${CFG_DIR}/funcoes.rc"

if [ -f ${CORES_RC} ]; then
. ${CORES_RC}
fi

# Lê as configurações do usuário
if [ -f ${CONFIGS_RC} ]; then
. ${CONFIGS_RC}
fi

# Carrega as configurações específicas ao sisteam de construção do GabiX
if [ -f ${GABIX_STUFF_RC} ]; then
. ${GABIX_STUFF_RC}
fi

# Configurações específicas para a plataforma ARM
if [ -f ${ARM_STUFF_RC} ]; then
. ${ARM_STUFF_RC}
fi

# Configurações específicas para x86 e x86_64
if [ -f ${GRUB_STUFF_RC} ]; then
. ${GRUB_STUFF_RC}
fi

# Configurações para a ferramenta de stripping de binários
if [ -f ${STRIP_STUFF_RC} ]; then
. ${STRIP_STUFF_RC}
fi

# Configurações para o kernel do GabiX
if [ -f ${KERNEL_STUFF_RC} ]; then
```

```
. ${KERNEL_STUFF_RC}
fi

# Configurações referentes aos idiomas e zonas do sistema
if [ -f ${LOCALES_STUFF_RC} ]; then
. ${LOCALES_STUFF_RC}
fi

# Configurações específicas para a construção do miniroot
if [ -f ${MINIROOT_STUFF_RC} ]; then
. ${MINIROOT_STUFF_RC}
fi

# Funções usadas por todo o sistema de construção e suas ferramentas
if [ -f ${FUNCOES_RC} ]; then
. ${FUNCOES_RC}
fi
```

A.2 Arquivo cores.rc

Define e ajusta as cores do Shell. Utilizado por todos os scripts de construção do GabiX.

```
#
# Configuracoes de ambiente shell para os scripts do GabiX
#
# Não execute este arquivo diretamente. Ele deve ser importado por cada script,
# quando necessario.
#
# Gabriel Marques
# snortt@gmail.com
#
# Sex Nov  4 14:48:15 BRST 2011
#

# --------------------------------
# Cores para o shell
RESET='\033[0m'
BRANCO='\033[39;1m'
VERMELHO='\033[31;1m'
VERDE='\033[32;1m'
AMARELO='\033[33;1m'
AZUL='\033[34;1m'
MAGENTA='\033[35;1m'
CIANO='\033[36;1m'
BRANCOB='\033[37;1m'

# -------------------------------
# Tema de cores
# Cores usadas nas mensagens de texto ao longo do boot
# Se desejar criar temas, é aqui que você deve modificar.
TEXTO="$AZUL"
FORTE="$BRANCOB"
ADERECO="$AZUL"
DEV="$MAGENTA"
MOD="$MAGENTA"
ACERTO="$VERDE"
```

```
AVISO="$AMARELO"
ERRO="$VERMELHO"
SEPARADOR="$AMARELO"
APONTADOR="$VERMELHO"
EXEC_SHELL="$VERMELHO"
```

A.3 Arquivo configs.rc

Utilizado por todos os scripts de construção do GabiX.

```
#
# Configuracoes globais para os scripts do GabiX
#
# Este arquivo contém as variáveis utilizadas por todos os
# scripts utilizados nas etapas de construção do GabiX Linux Live.
#
# Se precisar realizar alterações no ambiente de construção, utilize este arquivo.
#
# Não execute este arquivo diretamente. Ele deve ser importado por cada script,
# quando necessario.
#
# Gabriel Marques - snortt@gmail.com
# Sex Jun  3 09:57:42 BRT 2011
#


# ------------------------
# Configurações globais
# ------------------------
#
# Nome da versão do seu sistema Linux Live.
# Se você não gosta do nome GabiX ou tem outro nome melhor, use a variável abaixo
# para definir o nome de sua preferência. ;-)
#
# Nome que aparece na tela de boot e mensagens de boas vindas.
# Evite usar muitas palavras para nomear seu sistema!
NOMESTR="LNCC Linux"
# O mesmo que o anterior, porém para os arquivos do sistema. Evite espaços e acentos.
NOMESTR_SYS="LNCC-Linux"
# Nome do arquivo ISO que será criado.
ISOFILESTR="LNCC-Linux-i386.iso"

# Resolução da tela para o modo texto.
# Bits(Cores)  640x480  800x600  1024x768  1280x1024  1400x1050  1600x1200
# 8(256)       769      771      773       775       ---        ---
# 15(32K)      784      787      790       793       ---        ---
# 16(65K)      785      788      791       794        834        884
# 24(16M)      786      789      792       795       ---        ---
VGASTR="771"

# Aqui você pode especificar seu e-mail para contatos ;-)
EMAILSTR="snortt@gmail.com"

# Esta variável montará o cabeçalho do arquivo ISO.
PUBLISHERSTR="${NOMESTR} - ${EMAILSTR}"
```

```
# Configuração da rede
# Se for usar dhcp, não há necessidade de usar as variáveis do bloco
# Se for usar static, modifique as variáveis do bloco
#
## Ip dinâmico:
NETWORK_MODE="dhcp"

## Ip estático
#NETWORK_MODE="static"
#NETWORK_IP="146.134.35.176"
#NETWORK_MASK="255.255.255.0"
#NETWORK_DNS="146.134.8.160"
#NETWORK_GATEWAY="146.134.35.254"

# Se desejar iniciar a interface gráfica no boot, inicie a variável abaixo com "1"
USE_XORG="0"

# Se desejar usar o driver da Nvidia, inicie a varável abaixo com "1".
# USENVDRV="0"  - Não usa o driver Nvidia (proprietário)
# USENVDRV="1"  - Usa o driver Nvidia (proprietário)
USENVDRV="0"

# Caractere da barra de progresso, quando usada
# Aqui vc pode escolher o símbolo que prefere
# ver na tela quando uma barra de progresso for usada ;-)
# ".", "*", "=", "-", etc.
# Não confunda com a barra de progresso do "mksquashfs", que é hard coded ;-)
PROGRESSBAR="."

# ID do usuário que vai usar as ferramentas. Útil para quando não for usar o root.
# Note que alguns scripts ainda precisam de permissões de root (sudo cai bem ;-P )
FILES_USER_ID="1000"
FILES_GROUP_ID="1000"

# Tamanho da área de memória para o "tmpfs" usado pelo /dev
TMPFS_SIZE="64k"

# Modo de permissões para o /dev
TMPFS_MODE="0755"

# Diretório base de todo o ambiente
# ex: BASEDIR_GABIX="/home/biel/devel/GabiX-devel/"
# Este é mais legal! Ele funciona melhor por ser um caminho relativo ;-)
# ex: BASEDIR_GABIX="$PWD"
BASEDIR_GABIX="$PWD"

# ----------------------------------------
# Configurações de gravação das imagens ISO.
# Velocidade de gravação.
# Se for midia RW, melhor usar 4.
SPEED="4"

# Gravador.
REC_DEV="/dev/sr0"
```

```
# Programa de gravacao.
GRAVADOR="$(which wodim)"

# Opcoes de gravacao.
WODIM_OPTS="-dao -v"
```

A.4 Arquivo gabix_stuff.rc

Carrega as configurações específicas ao sisteam de construção do GabiX.

```
# Diretório base de todo o ambiente
# ex: BASEDIR_GABIX="/home/biel/devel/GabiX-devel/"
# Este é mais legal! Ele funciona melhor por ser um caminho relativo ;-)
# ex: BASEDIR_GABIX="$PWD"
BASEDIR_GABIX="$PWD"

# Nvidia driver dir
NVDRVDIR="nv_live"

# Diretorio que contem a raiz que dará origem à imagem ISO do gabix.
GABIXROOTDIR="${BASEDIR_GABIX}/gabix"

# Diretório de drivers na raiz da imagem ISO.
GABIXDRVDIR="${GABIXROOTDIR}/drv"

# Diretório na raiz do CD, onde está o driver da Nvidia.
GABIXNVDRVDIR="${GABIXDRVDIR}/${NVDRVDIR}"

# Aponte esta variável para onde estiver seu diretório "gabix/boot".
# Este diretório contém as de kernel utilizadas para iniciar o GabiX.
BOOTDIR="${GABIXROOTDIR}/boot"

# Vamos dizer onde estão os arquivos para carregar o driver da Nvidia
# Note que este local é um link simbólico para o verdadeiro driver, em
# ./kernel/nvidia_drv/nv_live_<arquitetura>, o que faz com que o
# montador da imagem SquashFS sempre combine as arquiteturas do sistema e
# do driver, de forma correta.
NVIDIAROOTDIR="${BASEDIR_GABIX}/${NVDRVDIR}"

# Aponte esta variável para onde estiver seu diretório "gabix/base".
# Este diretório contém as imagens SquashFS.
BASEDIR="${GABIXROOTDIR}/base"

# Nome do arquivo squashfs
# AVISO: Se for mudar, lembre-se de que o mesno nome é usado no init do miniroot!
SQUASHFILE="sistema"

# ------------------------------------------------
# Configurações de pacotes e arquivos de modelo.
# Diretorio com backups das configuracoes do sistema.
# Alguns arquivos deste diretório são utilizados como templates pelas
# diversas etapas de construção do GabiX.
CONFS_DIR="${BASEDIR_GABIX}/confs"

# Diretorio com arquivos empacotados que podem ser utilizados durante
# as etapas de construção do GabiX. Um deles é o tarball que guarda
```

```
# o conteúdo do "/dev" da imagem SquashFS. Antes de criar as imagens SquashFS,
# seu /dev/.{udev,init*} é compactado neste diretório e removido da raiz das imagens.
# Depois que as imagens forem construídas, seu /dev é restaurado.
# Uma imagem SquashFS não precisa ser criada com seu /dev/.udev e /dev/.init*
# porque estes são gerados dinamicamente durante o boot ;-).
CONFS_PKGS_DIR="${BASEDIR_GABIX}/confs/pacotes"

# Diretório com os modelos de configuração para a construção do nome do sistema
NOMESTR_SYS_FILES_TEMPLATES="${CONFS_DIR}/nomestr_sys_files"

# Arquivos para trocar o nome do sistema para NOMESTR e  NOMESTR_SYS
NOMESTR_SYS_FILE_HOSTS_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/hosts"
NOMESTR_SYS_FILE_HOSTNAME_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/hostname"
NOMESTR_SYS_FILE_RCLOCAL_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/rc.local"
NOMESTR_SYS_FILE_NOUVEAU_OFF_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/modprobe.d/\
    nouveau-off.conf"
NOMESTR_SYS_FILE_MOTD_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/motd"
NOMESTR_SYS_FILE_ISSUE_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/etc/issue"
NOMESTR_SYS_FILE_ROOTBASHRC_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/root/.bashrc"
NOMESTR_SYS_FILE_GABIXBASHRC_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/home/gabix/.bashrc"
NOMESTR_SYS_FILE_GABIXPROFILE_TEMPLATE="${NOMESTR_SYS_FILES_TEMPLATES}/home/gabix/.profile"

# Arquivos de destino com o nome do sistema para NOMESTR e  NOMESTR_SYS
NOMESTR_SYS_FILE_HOSTS="etc/hosts"
NOMESTR_SYS_FILE_HOSTNAME="etc/hostname"
NOMESTR_SYS_FILE_RCLOCAL="etc/rc.local"
NOMESTR_SYS_FILE_NOUVEAU_OFF="etc/modprobe.d/nouveau-off.conf"
NOMESTR_SYS_FILE_MOTD="etc/motd"
NOMESTR_SYS_FILE_ISSUE="etc/issue"
NOMESTR_SYS_FILE_ROOTBASHRC="root/.bashrc"
NOMESTR_SYS_FILE_GABIXBASHRC="home/gabix/.bashrc"
NOMESTR_SYS_FILE_GABIXPROFILE="home/gabix/.profile"

# ---------------------------------------
# Configurações para funções dos scripts.
# Diretório passado como argumento a ser processado. Ele é quem diz qual
# diretório deve ser processado por cada ferramenta.
# Será gabix, miniroot, etc. Depende de qual script/função utilizar a variável.
# Na dúvida, não altere esta variável.
ROOTDIR="$1"

# Usadas pelo script de criar SquashFS
MAKEIT=$(which mksquashfs)
UDEV_NET_CFG_FILE="${ROOTDIR}/etc/udev/rules.d/70-persistent-net.rules"

# Arquivo carregador do driver da Nvidia.
STR_NVIDIA_OFF="#\/cdrom\/drv\/nv_live\/carregar_nv_drv.sh"
STR_NVIDIA_ON="\/cdrom\/drv\/nv_live\/carregar_nv_drv.sh"
```

A.5 Arquivo arm_stuff.rc

Configurações específicas para a plataforma ARM.

```
# Arquivos referentes ao BOOT para o ARM.
# TODO: Mudar isso para um nome mais apropriado (pboot, mboot, etc.).
ARM_BOOT_CFG="${BASEDIR_GABIX}/confs/boot/armel"
```

A.6 Arquivo grub_stuff.rc

Configurações específicas para x86 e x86_64.

```
# Diretorio com os arquivos utilizados para gerar os menus do GRUB do Live CD.
CONFS_GRUB_DIR="${BASEDIR_GABIX}/confs/boot/grub"

# Aponte esta variável para o(s) arquivo(s) de configuração de boot do GabiX
# No caso do GRUB, faça:
GRUBCFG="${BOOTDIR}/grub/menu.lst"
GRUBCFG_LOCAL="${BOOTDIR}/grub/menu.lst.local"
GRUBCFG_TOOLS="${BOOTDIR}/grub/menu.lst.tools"
GRUBCFG_INSTALLDEB="${BOOTDIR}/grub/menu.lst.installdeb"

# Arquivos de modelo para a construção dos menus do GRUB
GRUBCFG_TEMPLATE="${CONFS_GRUB_DIR}/menu.lst.template"
GRUBCFG_TEMPLATE_LOCAL="${CONFS_GRUB_DIR}/menu.lst.local.template"
GRUBCFG_TEMPLATE_TOOLS="${CONFS_GRUB_DIR}/menu.lst.tools.template"
GRUBCFG_TEMPLATE_INSTALLDEB="${CONFS_GRUB_DIR}/menu.lst.installdeb.template"
```

A.7 Arquivo strip_stuff.rc

Configurações para a ferramenta de stripping de binários.

```
# Configurações para o strip (veja o LEIAME.txt antes de usar isso!)
# Estripa tudo!
#STRIP_OPTS="--strip-all \
    --strip-unneeded \
    --discard-all \
    --discard-locals \
    --verbose \
    -R .comment -R .note -R .note.ABI-tag"

# Estripa quase tudo ;-)
STRIP_OPTS="--strip-unneeded \
    --discard-all \
    --discard-locals \
    --verbose \
    -R .comment -R .note -R .note.ABI-tag"

STRIP_BIN_STR="executable"
STRIP_LIB_STR="shared object"
STRIP_FILE_TYPE="ELF"
STRIP_BIN_DIRS="bin sbin usr/bin usr/sbin usr/local/bin usr/local/sbin opt"
STRIP_LIB_DIRS="lib usr/lib usr/local/lib opt"
```

A.8 Arquivo kernel_stuff.rc

Configurações para o kernel do GabiX.

```
# Nome do arquivo que contem o miniroot com busybox e módulos do kernel.
# INITRD_NAME_GZ="initrd-2.6.38.2-gabix"
INITRD_NAME_GZ="initrd-$(basename $(stat -c %n \
    ${BASEDIR_GABIX}/gabix/boot/vmlinuz* 2> /dev/null ) \
    2> /dev/null | cut -d "-" -f2-3)"

# Versão do kernel atual em uso pelo sistema
# Não confunda esta variável com $kernel_version, usada apenas pela
```

```
# ferramenta de troca de kernels.
KERNELVERSION="$(basename $(stat -c %n \
    ${BASEDIR_GABIX}/gabix/boot/vmlinuz* 2> /dev/null) \
    2> /dev/null | cut -d "-" -f2-3)"

# ------------------------------------------------
# Configurações do empacotador/trocador de kernels.
# Modulos a serem copiados para o miniroot x86 e x86_64
MODULOS_PC=" \
aic79xx.ko aic7xxx.ko aic7xxx_old.ko aic94xx.ko ata_generic.ko ata_piix.ko \
atiixp.ko blk-cgroup.ko cdrom.ko cfq-iosched.ko cramfs.ko crc16.ko crc-itu-t.ko \
crc-t10dif.ko eata.ko ehci-hcd.ko exportfs.ko ext2.ko ext3.ko ext4.ko fat.ko \
floppy.ko hid.ko hwa-hc.ko ide-core.ko ide-generic.ko ide-cd_mod.ko \
ide-4drives.ko ide-cd_mod.ko ide-core.ko ide-cs.ko ide-gd_mod.ko ide-generic.ko \
ide-pci-generic.ko ide_platform.ko ide-pnp.ko ide-tape.ko \
i2c-tiny-usb.ko iscsi_boot_sysfs.ko iscsi_tcp.ko \
isofs.ko isp116x-hcd.ko isp1362-hcd.ko isp1760.ko jbd2.ko jbd.ko libahci.ko \
libata.ko loop.ko lzo_compress.ko mbcache.ko msdos.ko nls_base.ko nls_cp437.ko \
nls_cp850.ko nls_cp860.ko nls_iso8859-15.ko nls_iso8859-1.ko nls_utf8.ko \
ohci-hcd.ko oxu210hp-hcd.ko pata_ali.ko pata_amd.ko pata_artop.ko pata_atiixp.ko \
pata_atp867x.ko pata_cmd64x.ko pata_cs5520.ko pata_cs5530.ko pata_cs5536.ko \
pata_efar.ko pata_it821x.ko pata_jmicron.ko pata_marvell.ko pata_mpiix.ko \
pata_netcell.ko pata_ns87410.ko pata_ns87415.ko pata_oldpiix.ko pata_pcmcia.ko \
pata_pdc2027x.ko pata_pdc202xx_old.ko pata_piccolo.ko pata_rdc.ko pata_rz1000.ko \
pata_sc1200.ko pata_sch.ko pata_serverworks.ko pata_sil680.ko pata_sis.ko \
pata_triflex.ko pata_via.ko pcmcia.ko pcmcia_core.ko pcmcia_rsrc.ko piix.ko \
power_supply.ko r8a66597-hcd.ko rc-iodata-bctv7e.ko \
sata_inic162x.ko sata_mv.ko sata_nv.ko sata_promise.ko sata_qstor.ko sata_sil24.ko \
sata_sil.ko sata_sis.ko sata_svw.ko sata_sx4.ko sata_uli.ko sata_via.ko \
sata_vsc.ko scsi_debug.ko scsi_mod.ko scsi_tgt.ko scsi_transport_fc.ko \
scsi_transport_iscsi.ko scsi_transport_sas.ko scsi_transport_srp.ko scsi_wait_scan.ko \
sd_mod.ko sg.ko sl811_cs.ko sl811-hcd.ko squashfs.ko sr_mod.ko sx8.ko u132-hcd.ko \
uas.ko udf.ko uhci-hcd.ko ums-datafab.ko unionfs.ko usb-common.ko usbcore.ko \
usbhid.ko usb-storage.ko vfat.ko whci-hcd.ko xfs.ko xhci-hcd.ko yenta_socket.ko \
zlib_deflate.ko scsi_transport_spi.ko sym53c8xx.ko
"

# Modulos a serem copiados para o miniroot ARM
MODULOS_ARM=" \
aic79xx.ko aic7xxx.ko aic7xxx_old.ko aic94xx.ko cramfs.ko crc16.ko crc-itu-t.ko \
crc-t10dif.ko eata.ko ehci-hcd.ko exportfs.ko ext2.ko ext3.ko ext4.ko \
hid.ko i2c-tiny-usb.ko iscsi_boot_sysfs.ko iscsi_tcp.ko \
isp116x-hcd.ko isp1362-hcd.ko isp1760.ko jbd2.ko jbd.ko libahci.ko \
loop.ko lzo_compress.ko mbcache.ko msdos.ko nls_base.ko nls_cp437.ko \
nls_cp850.ko nls_cp860.ko nls_iso8859-15.ko nls_iso8859-1.ko nls_utf8.ko \
ohci-hcd.ko oxu210hp-hcd.ko power_supply.ko r8a66597-hcd.ko rc-iodata-bctv7e.ko \
    scsi_mod.ko scsi_tgt.ko scsi_transport_fc.ko scsi_transport_iscsi.ko \
    scsi_transport_sas.ko scsi_transport_srp.ko scsi_wait_scan.ko \
sd_mod.ko sg.ko sl811_cs.ko sl811-hcd.ko squashfs.ko sr_mod.ko sx8.ko u132-hcd.ko \
uas.ko udf.ko uhci-hcd.ko ums-datafab.ko unionfs.ko usb-common.ko usbcore.ko \
usbhid.ko usb-storage.ko vfat.ko whci-hcd.ko xfs.ko xhci-hcd.ko \
zlib_deflate.ko scsi_transport_spi.ko sym53c8xx.ko
"

# Diretório que contém os kernels do GabiX
```

```
KERNEL_DIR="${BASEDIR_GABIX}/kernel"

# Diretório onde os scripts que manipulam pacotes do kernel farão sua bagunça
KERNEL_TMP_AREA="${KERNEL_DIR}/tmp"

# Arquivo de controle da arquitetura configurada.
arch_control_file=".arch_control"
```

A.9 Arquivo locales_stuff.rc

Configurações referentes aos idiomas e zonas do sistema.

```
# Utilizado pelo script de limpeza de diretório pré-criação da imagem SquashFS
# Aqui estão os locales (suporte a idiomas) que serão removidos da distribuição,
# antes de criar o arquivo SquashFS
LOCALES_TO_WIPE="af am an ang ar ara as ast az az_IR be be@latin bg bn bn_IN br
                 bs byn ca ca_ES ca@valencia crh cs cy da de de_AT de_DE dz el
                 en en_AU en@boldquot en_CA en_GB en_NZ en@quot en@shaw en_US
                 en_US@piglatin eo es es_AR es_ES es_MX et et_EE eu fa fi fo fr
                 fr_FR ga gez gl gu haw he hi hr hu hy ia id io is it it_IT ja
                 ja_JP ka kk km kn ko kok ko_KR ku ky la lg li lt
                 lv mai mg mi mk ml mn mr ms mt my nb nb_NO nds ne nl nn no nso
                 oc or pa pl ps pt pt_PT rm ro ru ru_RU rw si sk sl so sq
                 sr sr@ije sr@latin sr@Latn sv sw ta te tg th ti tig tk tl tr tt
                 ug uk uk_UA ur ur_PK uz uz@cyrillic ve vi wa wal wo xh yi zh_CN
                 zh_HK zh_TW zu"

# Aqui estão os fuso-horários que serão removidos da distribuição, antes de criar o SquashFS
ZONES_TO_WIPE="Africa America Antarctica Arctic Asia Atlantic Australia Canada
   CET Chile CST6CDT Cuba EET Egypt Eire EST EST5EDT Etc Europe
   Factory GB GB-Eire GMT GMT0 GMT-0 GMT+0 Greenwich Hongkong HST
   Iceland Indian Iran iso3166.tab Israel Jamaica Japan Kwajalein
   Libya localtime MET Mexico Mideast MST MST7MDT Navajo NZ NZ-CHAT
   Pacific Poland Portugal PRC PST8PDT right ROC ROK
   Singapore SystemV Turkey UCT Universal US UTC WET W-SU Zulu"
```

A.10 Arquivo miniroot_stuff.rc

Configurações específicas para a construção do miniroot.

```
# ----------------------
# Configurações de sistema.
# Vamos dizer ao scritp onde está a raiz do miniroot (quem dará origem ao initrd).
MINIROOTDIR="${BASEDIR_GABIX}/miniroot"

# Arquivos usados para criar o init do miniroot
# x86 e X86_64
MINIROOT_INIT_TEMPLATE_GRUB="${CONFS_GRUB_DIR}/miniroot.init.template"
# ARM
MINIROOT_INIT_TEMPLATE_ARM="${ARM_BOOT_CFG}/miniroot.init.template.armel"

# Arquivos usados para criar /etc/funcoes.rc, de acordo com a arquitetura
# x86 e X86_64
MINIROOT_FUNCOES_TEMPLATE_GRUB="${CONFS_GRUB_DIR}/miniroot.funcoes.rc.template"
# ARM
MINIROOT_FUNCOES_TEMPLATE_ARM="${ARM_BOOT_CFG}/miniroot.funcoes.rc.template.armel"
```

```
# Arquivo de inicialização e funções do miniroot
MINIROOT_INIT="${MINIROOTDIR}/init"
MINIROOT_FUNCOES="${MINIROOTDIR}/etc/funcoes.rc"
```

A.11 Arquivo funcoes.rc

Define funções auxiliares e utilizado por todos os scripts de construção do GabiX. Este arquivo é criado a partir do *template* localizado em "confs/boot/{grub,armel}"

```
#
# Configuracoes de ambiente shell para os scripts do GabiX
#
# Não execute este arquivo diretamente. Ele deve ser importado por cada script,
# quando necessario.
#
# Gabriel Marques
# snortt@gmail.com
#
# Sex Nov  4 14:48:15 BRST 2011
#

# Carrega as definições globais
. includes.rc

# ----------------------------
# Funções auxiliares
# Funcao que imprime mensagens de sucesso
printOK() {
    echo -e "$@ ${ADERECO}[${ACERTO}OK${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens de erro
printError() {
    echo -e "$@ ${ADERECO}[${ERRO}Erro${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens de aviso
printWarn() {
    echo -e "$@ ${ADERECO}[${AVISO}Aviso${ADERECO}]${RESET}"
}

# Funcao que imprime mensagens
# Se receber 1 em $1, imprime mensagem extendida ("*** TEXTO ***").
# Do contrário, imprime mensagem comum ("TEXTO").
printMsg() {
    if [ "$1" == "1" ]; then
        shift
        echo -e "${ADERECO}*** ${TEXTO}$@ ${ADERECO}***${RESET}"
    else
        echo -e "${TEXTO}$@ ${RESET}"
    fi
}

# Detectar arquitetura configurada.
detect_arch() {
gabix_arch=$(stat miniroot 2> /dev/null | grep File | \
```

```
    tr -s " " | cut -d " " -f5 | sed s/\"//g)

case $gabix_arch in
"miniroot_x86")
printMsg "Arquitetura configurada: ${AVISO}32 bits"
echo ${gabix_arch#*_} > ${arch_control_file}
;;
"miniroot_x86_64")
printMsg "Arquitetura configurada: ${AVISO}64 bits"
echo ${gabix_arch#*_} > ${arch_control_file}
;;
"miniroot_armel")
printMsg "Arquitetura configurada: ${AVISO}ARM (armel)"
echo ${gabix_arch#*_} > ${arch_control_file}
;;
*)
printMsg "Sem arquitetura configurada"
printMsg "Execute ${AVISO}make help${TEXTO} para mais informações"
  echo "" > ${arch_control_file}
;;
esac
}
```

B Scripts do GabiX

B.1 *script* para ajustar pacotes do kernel

```
#!/bin/bash
#
# Script para converter os pacotes .deb do kernel em tarball para o GabiX
#
# Este script ajusta os links simbolicos:
#  /lib/modules/$(uname -r)/build -> /usr/src/linux-headers-$(uname -r)
#  /lib/modules/$(uname -r)/source -> /usr/src/linux-headers-$(uname -r)
#  /usr/src/linux -> ./linux-headers-$(uname -r)
#
# Gabriel Marques - snortt@gmail.com
# Sex Jun  3 17:09:42 BRT 2011
#

# Carrega as definições globais
. includes.rc

if [ $# -lt 1 ]; then
    # Passe os arquivos que desejar (image, headers, firmware, libc-dev)
    printMsg "Uso: ${0##*/} <linux-image-x.y.z.deb> [<...>]"
exit 201
elif [ ! -L gabix ]; then
    printError "Não achei diretório de destino."
    printWarn "Por favor, execute make config_32 ou config_64"
    exit 202
else
# Note que este procedimento independe da arquitetura selecionada
# para o GabiX. Fornecer um kernel para uma arquitetura diferente
```

```
# da configurada atualmente é normal.
    printMsg "----------------------"
    printMsg "Detectando arquitetura"
    if file $1 | grep -i i386 > /dev/null 2>&1; then
        ARCH="x86"
        printOK "${ARCH} : "
    elif file $1 | grep -i amd64 > /dev/null 2>&1; then
        ARCH="x86_64"
        printOK "${ARCH} : "
    elif file $1 | grep -i armel > /dev/null 2>&1; then
        ARCH="armel"
        printOK "${ARCH} : "
else
printError "Arquitetura não suportada (ainda)."
exit 203
    fi

    printMsg "-------------------------------------------"
    printMsg "Extraindo conteudo dos pacotes do kernel..."
    dpkg -x ${1} ${KERNEL_TMP_AREA} && printOK "${1##*/} : " \
        || printError "${1##*/} : "
    dpkg -x ${2} ${KERNEL_TMP_AREA} && printOK "${2##*/} : " \
        || printError "${2##*/} : "

  # Nao confunda esta variavel com KERNELVERSION, do arquivo
  # configs.rc!
    # Esta aqui sera utilizada para realizar a troca de kernel!
    # Aquela aponta para
    # o kernel que ja esta instalado, para criar o menu do
    # gerenciador de boot! ;-)
    kernel_version=$(basename $(stat -c %n ${KERNEL_TMP_AREA}/boot/vmlinuz*) \
        | cut -d "-" -f2-3)
    printMsg "------------------------------------------------------"
    printMsg "Ajustando links e diretórios do kernel ${AVISO}${kernel_version}."
(cd ${KERNEL_TMP_AREA}; \
  rm lib/modules/${kernel_version}/build lib/modules/${kernel_version}/source \
  2> /dev/null) && printOK "Limpo : " || \
  printWarn "Limpo : Alguns arquivos faltando. Sem problemas! : "

(cd ${KERNEL_TMP_AREA}; ln -s /usr/src/linux-headers-${kernel_version} \
  lib/modules/${kernel_version}/source 2> /dev/null) && \
  printOK "Source : " || \
  printWarn "Source : Arquivo ja existe? Sem problemas! : "

(cd ${KERNEL_TMP_AREA}; ln -s /usr/src/linux-headers-${kernel_version} \
  lib/modules/${kernel_version}/build 2> /dev/null) && printOK "Build : " || \
  printWarn "Build : Arquivo ja existe? Sem problemas! : "

(cd ${KERNEL_TMP_AREA}/usr/src/; ln -s linux-headers-${kernel_version} \
  linux 2> /dev/null) && printOK "Linux : " || \
  printWarn "Linux : Arquivo ja existe? Sem problemas! : "

    printMsg "------------------------------------------------------------"
    printMsg "Empacotando novo kernel. Por favor aguarde alguns instantes."
    (
```

```
     cd $KERNEL_TMP_AREA
    tar cjf ${KERNEL_DIR}/linux-${kernel_version}.tar.bz2 * && printOK \
      "${kernel_version} : " || printError "${kernel_version} : "
)
    printMsg "Removendo arquivos temporários."
rm -r ${KERNEL_TMP_AREA}/* && printOK "$(basename ${KERNEL_TMP_AREA}) : "

chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${KERNEL_DIR} && \
  printOK "Posse de ${NOMESTR} Kernel : " || \
  printError "Posse de ${NOMESTR} Kernel : "

chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${GABIXROOTDIR} && \
  printOK "Posse de ${NOMESTR} BASE : " || \
  printError "Posse de ${NOMESTR} BASE : "

    chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${MINIROOTDIR} && \
      printOK "Posse de ${NOMESTR} Miniroot : " || \
      printError "Posse de ${NOMESTR} Miniroot : "
fi
```

B.2 *script* para troca de kernels

```
#!/bin/bash
#
# Este script troca o kernel do miniroot (initrd)
#
# Gabriel Marques - snortt@gmail.com
# Qui Mai  5 15:21:00 BRT 2011
#
# Este script precisa executar em uma maquina com a mesma versao do kernel
# que ele for instalar no miniroot e no gabix
#
# Lembre-se de usar a linha de comando
# "make-kpkg --initrd kernel_image kernel_headers"
# para gerar pacotes.deb com os arquivos do kernel.
#

# Carrega as definições globais
. includes.rc

#depscript="deps"

if [ "$(id -u)" -ne "0" ]; then
    printMsg "Voce precisa ser root"
    exit 201
fi

if [ $# -ne 1 ]; then
    printMsg "Uso: ${0##*/} <kernel_ajustado_para_gabix>"
exit 202
fi

if [ ! -L miniroot ]; then
    printError "Não achei diretório do miniroot."
    printWarn "Por favor, execute make config_32 ou config_64"
    exit 203
```

```
fi

if [ ! -f "$1" -o ! -d ${KERNEL_TMP_AREA} ]; then
    printMsg "${ERRO}Erro${TEXTO}. Nao achei ${DEV}$1 ${TEXTO}ou \
        ${DEV}${KERNEL_TMP_AREA}"
exit 203
else
    printMsg " "
    printMsg "----------------------"
    printMsg "Removendo kernel atual"
    printMsg "----------------------"
    # Caso algum arquivo esteja faltando e confunda o 'rm', vamos
    # imprimir apenas um aviso. Isso não significa uma falha!
    rm ${GABIXROOTDIR}/boot/{vmlinuz,initrd,config,System}* \
        2> /dev/null && printOK "${NOMESTR} BOOT : " || \
        printWarn "BOOT : Nao achei alguns arquivos. Sem problemas! : "

    rm ${GABIXROOTDIR}/base/{modulos,headers,firmware} 2> /dev/null && \
        printOK "${NOMESTR} BASE : " || \
        printWarn "BASE : Nao achei alguns arquivos. Sem problemas! : "

    rm -r ${MINIROOTDIR}/lib/modules/* 2> /dev/null && printOK "${NOMESTR} \
        : Miniroot : modulos : " || printWarn "Miniroot : modulos : \
        Nao achei alguns arquivos. Sem problemas! : "

    printMsg " "
    printMsg "-----------------------------------------"
    printMsg "Extraindo conteudo do pacote do novo kernel"
    printMsg "-----------------------------------------"
    # Aqui é importante verificar se o tar falhou! Isso sim pode atrapalhar tudo!
    tar xjf ${1} -C ${KERNEL_TMP_AREA} && printOK "${1##*/} : " \
        || printError "${1##*/} : "

    printMsg " "
    # Nao confunda esta variavel com KERNELVERSION, do arquivo configs_env.sh!
    # Esta aqui sera utilizada para realizar a troca de kernel! Aquela aponta para
    # o kernel que ja esta instalado, para criar o menu do gerenciador de boot! ;-)
    kernel_version=$(basename $(stat -c %n ${KERNEL_TMP_AREA}/boot/vmlinuz*) \
        | cut -d "-" -f2-3)

    printMsg "--------------------------------------"
    printMsg "Instalando novo kernel: ${DEV}$kernel_version"
    printMsg "--------------------------------------"
    cp ${KERNEL_TMP_AREA}/boot/{vmlinuz,config,System}* "${GABIXROOTDIR}/boot" \
        2> /dev/null && printOK "${NOMESTR} BOOT : " || \
        printError "${NOMESTR} BOOT : "
    (
        (cd ${KERNEL_TMP_AREA}/lib/ && mksquashfs modules \
            ${GABIXROOTDIR}/base/modulos \
            > /dev/null 2>&1) && \
            printOK "${NOMESTR} BASE : modulos : " || \
            printError "${NOMESTR} BASE : modulos : "
    )
    (
        (cd ${KERNEL_TMP_AREA}/lib/ && mksquashfs firmware \
```

```
            ${GABIXROOTDIR}/base/firmware \
            > /dev/null 2>&1) && \
            printOK "${NOMESTR} BOOT : firmware : " || \
            printError "${NOMESTR} BOOT : firmware : "
    )
    (
        (cd ${KERNEL_TMP_AREA}/usr/ && mksquashfs src ${GABIXROOTDIR}/base/headers \
            > /dev/null 2>&1 ) && \
            printOK "${NOMESTR} BOOT : headers : " || \
            printError "${NOMESTR} BOOT : headers : "

  if [ -d ${KERNEL_TMP_AREA}/usr/include ]; then
(cd ${KERNEL_TMP_AREA}/usr/ && mksquashfs include \
    ${GABIXROOTDIR}/base/includes > /dev/null 2>&1 ) && \
            printOK "${NOMESTR} BASE : includes : " || \
            printError "${NOMESTR} BASE : includes : "
  else
  printWarn "Este kernel nao possui diretorio include"
  fi
    )

    printMsg " "
    printMsg "Criando diretorio de modulos do miniroot para a versao \
        ${DEV}$kernel_version"
   mkdir ${MINIROOTDIR}/lib/modules/${kernel_version} 2> /dev/null && \
      printOK || printError

    # ====================
# Detectar arquitetura
# ====================
# Antes de prosseguir, vamos detectar a arquitetura.
    # ARM precisa de tratamentos diferenciados ;-)
# Uma vez detectada a arquitetura, basta editar os templates apropriados
# (veja confs/boot/) e, se necessário, o arquivo "funções" dentro
# de miniroot/etc/.
    printMsg " "
detect_arch
arch=$(cat ${arch_control_file})

case ${arch} in
"armel")
MODULOS="${MODULOS_ARM}"
;;
"x86"|"x86_64")
MODULOS="${MODULOS_PC}"
;;
*) printMsg "Problema ao detectar arquitetura para construir miniroot"
   printError "Isso é um erro sério! Abortado."
   exit 203
   ;;
esac

    printMsg " "
    printMsg "Copiando modulos para o miniroot"
    (
```

```
    for modulo in $MODULOS
    do
find ${KERNEL_TMP_AREA} -name "$modulo" -exec cp -f {} \
    ${MINIROOTDIR}/lib/modules/${kernel_version} \; 2> /dev/null && \
    echo -en "${APONTADOR}${PROGRESSBAR}"
     done
    ) && printOK || printError

    printMsg " "
# O init do miniroot ja faz isso
# Se for utilizar este bloco, descomente a variavel ‘depscript‘
    #printMsg "Gerando dependencias no miniroot"
    #(
#    echo "mount /proc" > ${MINIROOTDIR}/${depscript}
#    echo "depmod 2> /dev/null" >> ${MINIROOTDIR}/${depscript}
#    echo "umount /proc" >> ${MINIROOTDIR}/${depscript}
#
#    chmod +x ${MINIROOTDIR}/${depscript}
#    chroot ${MINIROOTDIR} /${depscript}
#    rm ${MINIROOTDIR}/${depscript}
    #) && printOK || printError

    printMsg "------------------------------"
    printMsg "Removendo arquivos temporarios"
    printMsg "------------------------------"
    rm -r ${KERNEL_TMP_AREA}/* && printOK || printError "Remova manualmente : "

    printMsg " "
    printMsg "--------------------------------"
    printMsg "Gerando miniroot com novo kernel"
    printMsg "--------------------------------"
./criar_miniroot.sh miniroot

    echo ""
chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${GABIXROOTDIR} && \
    printOK "Posse de ${NOMESTR} BASE : " || \
    printError "Posse de ${NOMESTR} BASE : "

    chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${MINIROOTDIR} && \
      printOK "Posse de ${NOMESTR} Miniroot : " || \
      printError "Posse de ${NOMESTR} Miniroot : "
fi
```

B.3 *script* para criar imagens SquashFS

```
#!/bin/bash
#
# criar_squashfs.sh
#
# Script para criar imagem squashfs do debian bootstrapped
#
# Gabriel Marques
# snortt@gmail.com
#
# Sex Mai  1 17:31:09 BRT 2011
```

```
#

# Carrega as definições globais
. includes.rc

if [ ! -x "$MAKEIT" ]; then
    printMsg "Voce precisa instalar ${ERRO}squashfs-tools"
    exit 201
fi

if [ "$(id -u)" -ne "0" ]; then
    printMsg "Voce precisa ser root"
    exit 202
fi

if [ $# -lt 1 ]; then
    printMsg "Uso: ${0##*/} <debian_root_dir>"
exit 203
fi

if [ ! -d $ROOTDIR ]; then
    # Vamos só disparar um aviso, pois o miniroot funciona, mesmo
    # sem roo-NFS no caso de um make all
    printWarn "Nao achei ${DEV}$ROOTDIR"
exit 204
else
    printMsg " "
    printMsg "---------------------------------"
    printMsg "Removendo arquivo ${MAGENTA}${SQUASHFILE} ${TEXTO}anterior"
    printMsg "---------------------------------"
    # Caso o arquivo já tenha sido removido, vamos imprimir apenas um aviso.
    if [ -f "${BASEDIR}/${SQUASHFILE}" ]; then
        rm "${BASEDIR}/${SQUASHFILE}" 2> /dev/null
        printOK
    else
        printWarn "${MAGENTA}${SQUASHFILE} ${RESET}anterior nao encontrado. \
          Sem problemas! : "
    fi

  # Desligamento do /dev da imagem SquashFS. Deixe o kernel e o udev lidarem
   # com isso na hora que eles carregarem live ;-)
    printMsg " "
    printMsg "-----------------------------------"
    printMsg "Desativando dispositivos em ${DEV}${ROOTDIR}"
    printMsg "-----------------------------------"
   (tar cjf ${CONFS_PKGS_DIR}/dev_stuff.tar.bz2 ${ROOTDIR}/dev/ && \
    rm -rf ${ROOTDIR}/dev/.{udev,initram}*) && printOK || printError

   # Ajustes nos arquivos de ambiente do sistema
    printMsg " "
    printMsg "----------------------------------------"
    printMsg "Ajustando nome do sistema para ${DEV}${NOMESTR_SYS}"
    printMsg "----------------------------------------"
    cat $NOMESTR_SYS_FILE_HOSTNAME_TEMPLATE | \
      sed s/__NOMESTR_SYS__/"${NOMESTR_SYS}"/g > \
```

```
    ${ROOTDIR}/${NOMESTR_SYS_FILE_HOSTNAME} && \
    printOK "hostname :: " || \
    printError printOK "hostname :: "

  cat $NOMESTR_SYS_FILE_HOSTS_TEMPLATE | \
    sed s/__NOMESTR_SYS__/"${NOMESTR_SYS}"/g > \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_HOSTS} && \
    printOK "hosts :: " || \
    printError "hosts :: "

  cat $NOMESTR_SYS_FILE_MOTD_TEMPLATE | \
    sed s/__NOMESTR__/"${NOMESTR_SYS}"/g > \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_MOTD} && \
    printOK "motd :: " || \
    printError "motd :: "

  cat $NOMESTR_SYS_FILE_ISSUE_TEMPLATE | \
    sed s/__NOMESTR__/"${NOMESTR_SYS}"/g > \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_ISSUE} && \
    printOK "issue :: " || \
    printError "issue :: "

 # Ajustes no arquivo .bashrc do root.
  cat $NOMESTR_SYS_FILE_ROOTBASHRC_TEMPLATE | \
    sed s/__NOMESTR_SYS__/"${NOMESTR_SYS}"/g > \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_ROOTBASHRC} && \
    printOK "root.bashrc :: " || \
    printError "root.bashrc :: "

 # Ajustes no arquivo .bashrc do usuário gabix.
  cat $NOMESTR_SYS_FILE_GABIXBASHRC_TEMPLATE | \
    sed s/__NOMESTR_SYS__/"${NOMESTR_SYS}"/g > \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_GABIXBASHRC} && \
    printOK "gabix.bashrc :: " || \
    printError "gabix.bashrc :: "

 # Ajustes no .profile do usuário gabix.
  cp $NOMESTR_SYS_FILE_GABIXPROFILE_TEMPLATE \
    ${ROOTDIR}/${NOMESTR_SYS_FILE_GABIXPROFILE} && \
    printOK "gabix.profile :: " || \
    printError "gabix.profile :: "

# Aqui trataremos os recursos adicionais, controlados via rc.local,
# pós-boot do Debian.
 printMsg " "
 printMsg "Configurando ${DEV}rc.local${TEXTO}"

 # --------------------------------------
 # Verificando suporte por drivers Nvidia.
 case "$USENVDRV" in
 "0")
 printMsg "Desligando o driver ${AVISO}Nvidia ${TEXTO}."
 rm -r ${GABIXNVIDIAROOTDIR}/*  > /dev/null 2>&1 && printOK || \
 printWarn "Existem arquivos la? : "
 printMsg "Nao tocando em rc.local da imagem SquashFS"
```

```
 printMsg "Ligando carregamento do driver ${AVISO}Nouveau${TEXTO}"
 rm ${ROOTDIR}/${NOMESTR_SYS_FILE_NOUVEAU_OFF} > /dev/null 2>&1 && \
 printOK "nouveau on :: " || \
 printWarn "nouveau on :: Arquivo de bloqueio existe? :: "
 ;;

"1")
 printMsg "Habilitando uso do driver ${AVISO}Nvidia${TEXTO}"
 cat ${NOMESTR_SYS_FILE_RCLOCAL_TEMPLATE} | \
 sed s/"${STR_NVIDIA_OFF}/${STR_NVIDIA_ON}"/g > \
 ${ROOTDIR}/${NOMESTR_SYS_FILE_RCLOCAL} && \
 printOK "rc.local :: ${AVISO}Nvidia ${RESET}:: " || \
 printError "rc.local :: ${AVISO}Nvidia ${RESET} :: "

 printMsg "Desligando carregamento do driver ${AVISO}Nouveau${TEXTO}"
 cp $NOMESTR_SYS_FILE_NOUVEAU_OFF_TEMPLATE \
 ${ROOTDIR}/${NOMESTR_SYS_FILE_NOUVEAU_OFF} > /dev/null 2>&1 && \
 printOK "nouveau off :: ${AVISO}Nvidia ${RESET}:: " || \
 printError "nouveau off :: ${AVISO}Nvidia ${RESET} :: "

 # Aqui poderíamos usar links absolutos, mas vamos apenas copiar os arquivos,
 # partindo do princípio de que o desenvolvedor poderia zonear algum
 # arquivo por engano.
 printMsg "Inserindo o driver na raiz da imagem"
 cp -ar ${NVIDIAROOTDIR}/* ${GABIXNVIDIAROOTDIR}/ > /dev/null 2>&1 && \
 printOK || printWarn "Faltou copiar algum arquivo? : "
 ;;

 *)
 printWarn "Valor desconhecido para USENVDRV : ${USENVDRV}"
 printMsg "Verifique o arquivo configs.rc"

 ;;
 esac

 printMsg " "
 printMsg "-----------------------------------------------"
 printMsg "Desligando interface de rede no arquivo do udev"
 printMsg "-----------------------------------------------"
 printMsg "Importante para evitar confusao na hora de subir a interface de rede."
 (
  head -6 ${UDEV_NET_CFG_FILE} > /tmp/criando_squash_file
 mv /tmp/criando_squash_file ${UDEV_NET_CFG_FILE}
 ) && printOK || printWarn "Arquivo existe? : "

  printMsg " "
  printMsg "-----------------------"
  printMsg "Criando imagem squashfs"
  printMsg "-----------------------"
  printMsg "Aguarde alguns instantes."
  $MAKEIT "${ROOTDIR}/" "${BASEDIR}/${SQUASHFILE}" &> \
  /dev/null && printOK || printError

  printMsg " "
```

```
    printMsg "Ajustando a posse do sistema para ${AVISO}UID:${FILES_USER_ID}"
  chown ${FILES_USER_ID}.${FILES_GROUP_ID} ${BASEDIR}/* && \
    printOK || printError

    printMsg " "
    printMsg "-------------------------------------"
    printMsg "Reativando dispositivos em ${DEV}${ROOTDIR}"
    printMsg "-------------------------------------"
   tar xjf ${CONFS_PKGS_DIR}/dev_stuff.tar.bz2 -C . && \
    printOK || printError

    printMsg " "
  chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${GABIXROOTDIR} && \
    printOK "Posse de ${NOMESTR} : " || \
    printError "Posse de ${NOMESTR} : "
fi
```

B.4 *script* para criar o miniroot

```
#!/bin/bash
#
# criar_miniroot.sh
#
# Este script cria uma imagem miniroot compactada.
#
# Gabriel Marques - snortt@gmail.com
# Sex Dez  4 14:46:58 BRST 2009

# Carrega as definições globais
. includes.rc

if [ $# -lt 1 ]; then
    printMsg "Uso: ${0##*/} <miniroot_dir>"
exit 201
fi

if [ ! -d $ROOTDIR ]; then
    printMsg "${VERMELHO}Erro${TEXTO}. Nao achei ${DEV}$ROOTDIR"
exit 202
else
    printMsg " "
    printMsg "Removendo arquivo ${DEV}${INITRD_NAME_GZ} ${TEXTO}anterior"
if [ -f ${BOOTDIR}/${INITRD_NAME_GZ} ]; then
        rm ${BOOTDIR}/${INITRD_NAME_GZ}
        printOK
    else
        printWarn "${DEV}${INITRD_NAME_GZ} ${RESET}anterior nao encontrado. Sem problemas! : "
    fi

# ====================
# Detectar arquitetura
# ====================
# Antes de prosseguir, vamos detectar a arquitetura.
    # ARM precisa de tratamentos diferenciados ;-)
# Uma vez detectada a arquitetura, basta editar os templates apropriados
# (veja confs/boot/) e, se necessário, o arquivo "funções" dentro
```

```
# de miniroot/etc/.
detect_arch
arch=$(cat ${arch_control_file})

case ${arch} in
"armel")
MINIROOT_INIT_TEMPLATE="${MINIROOT_INIT_TEMPLATE_ARM}"
;;
"x86"|"x86_64")
MINIROOT_INIT_TEMPLATE="${MINIROOT_INIT_TEMPLATE_GRUB}"
;;
*) printMsg "Problema ao detectar arquitetura para construir miniroot"
   printError "Isso é um erro sério! Abortado."
   exit 203
   ;;
esac

cd $ROOTDIR

# ====================
# AJUSTES NO MINIROOT
# ====================
#
# Arquivo /init
    printMsg " "
printMsg "Criando arquivo init no miniroot"
    cat ${MINIROOT_INIT_TEMPLATE} > ${MINIROOT_INIT} && \
printOK "init :: ${MINIROOT_INIT#*miniroot} :: " || \
printError "init :: ${MINIROOT_INIT#*miniroot} :: "

    printMsg "Ajustando o nome para ${DEV}$NOMESTR"
    sed -i s/"__NOMESTR__/${NOMESTR}"/g ${MINIROOT_INIT} && \
printOK "init :: $NOMESTR :: " || printError "init :: $NOMESTR :: "

printMsg " "
printMsg "Ajustando tamanho da área de memória para TMPFS usada pelo /dev"
    sed -i s/"__TMPFS_SIZE__/${TMPFS_SIZE}"/g ${MINIROOT_INIT} && \
printOK "init :: ${TMPFS_SIZE} :: " || printError "init :: $TMPFS_SIZE :: "

printMsg " "
printMsg "Ajustando permissões do diretório /dev"
    sed -i s/"__TMPFS_MODE__/${TMPFS_MODE}"/g ${MINIROOT_INIT} && \
printOK "init :: ${TMPFS_MODE} :: " || printError "init :: $TMPFS_MODE :: "

printMsg " "
printMsg "Aplicando permissões no arquivo init"
    chmod +x ${MINIROOT_INIT} && \
printOK "Permissao exec :: init :: $NOMESTR :: " || \
printError "Permissao exec :: init :: $NOMESTR :: "

# Arquivo /etc/funcoes.rc
printMsg " "
printMsg "Criando arquivo funcoes.rc no miniroot/etc/"
    cat ${MINIROOT_FUNCOES_TEMPLATE} > ${MINIROOT_FUNCOES} && \
printOK "init :: ${MINIROOT_FUNCOES#*miniroot} :: " || \
```

```
    printError "init :: ${MINIROOT_FUNCOES#*miniroot} :: "

# ================================================
# Criação da imagem initramfs, a partir do miniroot
# ================================================
    printMsg " "
    printMsg "Criando imagem initramfs em ${DEV}${INITRD_NAME_GZ}"
find . | cpio --quiet -o -H newc | gzip > ../${INITRD_NAME_GZ} && \
    printOK || printError
cd ../

    printMsg " "
    printMsg "Movendo ${DEV}${INITRD_NAME_GZ} ${TEXTO}para \
      ${DEV}gabix/boot/${INITRD_NAME_GZ}"
  mv ${INITRD_NAME_GZ} ${BOOTDIR}/ && printOK || printError

    printMsg " "
    printMsg "Construindo menu de inicializacao do sistema."
   cat ${GRUBCFG_TEMPLATE} | \
     sed s/__VERSAO__/$KERNELVERSION/g > ${GRUBCFG} && \
    printOK "${NOMESTR} Menu :: " || \
    printError "${NOMESTR} Menu :: "

  cat ${GRUBCFG_TEMPLATE_LOCAL} | \
    sed s/__VERSAO__/$KERNELVERSION/g > ${GRUBCFG_LOCAL} && \
      printOK "${NOMESTR} Local :: " || \
      printError "${NOMESTR} Local :: "

  cat ${GRUBCFG_TEMPLATE_TOOLS} | \
    sed s/__VERSAO__/$KERNELVERSION/g > ${GRUBCFG_TOOLS} && \
    printOK "${NOMESTR} Tools :: " || \
    printError "${NOMESTR} Tools :: "

    cat ${GRUBCFG_TEMPLATE_INSTALLDEB} | \
      sed s/__VERSAO__/$KERNELVERSION/g > ${GRUBCFG_INSTALLDEB} && \
      printOK "${NOMESTR} Installdeb :: " || \
      printError "${NOMESTR} Installdeb :: "

    printMsg " "
printMsg "Ajustando configuracoes de rede"
if [ "$NETWORK_MODE" == "static" ]; then
sed -i s/__NETMODE__/"${NETWORK_MODE}"/g ${GRUBCFG} && \
    printOK "$NETWORK_MODE ::   Menu :: " || \
    printError "$NETWORK_MODE :: Menu :: "

sed -i s/__NETIP__/"${NETWORK_IP}"/g ${GRUBCFG} && \
    printOK "$NETWORK_IP :: Menu :: " || \
    printError "$NETWORK_IP :: Menu :: "

sed -i s/__NETMASK__/"${NETWORK_MASK}"/g ${GRUBCFG} &&
     printOK "$NETWORK_MASK :: Menu :: " || \
     printError "$NETWORK_MASK :: Menu :: "

sed -i s/__NETDNS__/"${NETWORK_DNS}"/g ${GRUBCFG} && \
     printOK "$NETWORK_DNS :: Menu :: " || \
```

```
     printError "$NETWORK_DNS :: Menu :: "

sed -i s/__NETGTWY__/"${NETWORK_GATEWAY}"/g ${GRUBCFG} && \
     printOK "$NETWORK_GATEWAY :: Menu :: " || \
     printError "$NETWORK_GATEWAY :: Menu :: "
else
sed -i s/__NETMODE__/"${NETWORK_MODE}"/g ${GRUBCFG} && \
     printOK "$NETWORK_MODE :: Menu :: " || \
     printError "$NETWORK_MODE :: Menu :: "

sed -i s/__NETIP__/" "/g ${GRUBCFG} && \
     printOK "IP (null) :: Menu :: " || \
     printError "IP (null) :: Menu :: "

sed -i s/__NETMASK__/" "/g ${GRUBCFG} && \
     printOK "Mask (null) :: Menu :: " || \
     printError "Mask (null) :: Menu :: "

sed -i s/__NETDNS__/" "/g ${GRUBCFG} && \
     printOK "DNS (null) :: Menu :: " || \
     printError "DNS (null) :: Menu :: "

sed -i s/__NETGTWY__/" "/g ${GRUBCFG} && \
     printOK "Gateway (null) :: Menu :: " || \
     printError "Gateway (null) :: Menu :: "

sed -i s/\ \ */\ /g ${GRUBCFG} && \
     printOK "Ajuste de espacos em branco :: Menu :: " || \
     printError "Ajuste de espacos em branco ::Menu :: "
fi

    printMsg " "
    printMsg "Ajustando o nome para ${AVISO}$NOMESTR \
      ${TEXTO}no menu do sistema."

   sed -i s/__NOMESTR__/"${NOMESTR}"/g ${GRUBCFG} && \
     printOK "$NOMESTR Menu :: " || \
     printError "$NOMESTR Menu :: "

    sed -i s/__NOMESTR__/"${NOMESTR}"/g ${GRUBCFG_LOCAL} && \
      printOK "$NOMESTR Local :: " || \
      printError "$NOMESTR Local :: "

  sed -i s/__NOMESTR__/"${NOMESTR}"/g ${GRUBCFG_TOOLS} && \
    printOK "$NOMESTR Tools :: " || \
    printError "$NOMESTR Tools :: "

    sed -i s/__NOMESTR__/"${NOMESTR}"/g ${GRUBCFG_INSTALLDEB} && \
      printOK "$NOMESTR Installdeb :: " || \
      printError "$NOMESTR Installdeb :: "

    printMsg " "
    printMsg "Ajustando resolucao inicial para \
      ${AVISO}$VGASTR ${TEXTO}no menu do sistema."
```

```
   sed -i s/__VGASTR__/"${VGASTR}"/g ${GRUBCFG} && \
     printOK "$VGASTR Menu :: " || \
     printError "$VGASTR Menu :: "

    sed -i s/__VGASTR__/"${VGASTR}"/g ${GRUBCFG_LOCAL} && \
      printOK "$VGASTR Local :: " || \
      printError "$VGASTR Local :: "

  sed -i s/__VGASTR__/"${VGASTR}"/g ${GRUBCFG_TOOLS} && \
    printOK "$VGASTR Tools :: " || \
    printError "$VGASTR Tools :: "

    sed -i s/__VGASTR__/"${VGASTR}"/g ${GRUBCFG_INSTALLDEB} && \
      printOK "$VGASTR Installdeb :: " || \
      printError "$VGASTR Installdeb :: "

    printMsg " "
   chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${GABIXROOTDIR} && \
     printOK "Posse de ${NOMESTR} BASE : " || \
     printError "Posse de ${NOMESTR} BASE : "

    chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${MINIROOTDIR} && \
      printOK "Posse de ${NOMESTR} Miniroot : " || \
      printError "Posse de ${NOMESTR} Miniroot : "
fi
```

B.5 *script* para criar o ISO

```
#!/bin/bash
#
# make-iso
#
# Este script cria uma imagem ISO do GabiX
#
# Gabriel Marques - snortt@gmail.com
# Seg Jun  6 12:00:02 BRT 2011
#

# Carrega as definições globais
. includes.rc

if [ $# -ne 1 ]; then
  printMsg "Uso: ${0##*/} <gabix_cdroot_dir>"
exit 201
else
  printMsg " "
  printMsg "Gerando arquivo ISO para ${DEV}${NOMESTR}"
genisoimage -l -r -J -v -publisher "${PUBLISHERSTR}" \
  -preparer "${PUBLISHERSTR}" -V "${PUBLISHERSTR}" \
    -iso-level 4 -R -U -hide-rr-moved \
    -cache-inodes -no-bak -pad -no-emul-boot -boot-info-table \
-b boot/grub/iso9660_stage1_5 -c boot/boot.cat -boot-load-size 4 \
    -o ${ISOFILESTR} $1/ > /dev/null 2>&1 && \
    printOK "${NOMESTR} : ISO : " || \
    printError "${NOMESTR} : ISO : "
```

```
  chown -R ${FILES_USER_ID}.${FILES_GROUP_ID} ${ISOFILESTR} && \
    printOK "Posse de ${NOMESTR} ISO : " || \
    printError "Posse de ${NOMESTR} ISO : "
fi
```

C *Patch* para o logotipo no kernel

```
diff -ruN linux-2.6.38.2.vanilla/drivers/video/logo/Kconfig \
        linux-2.6.38.2-gabix/drivers/video/logo/Kconfig
--- linux-2.6.38.2.vanilla/drivers/video/logo/Kconfig 2011-03-27 \
        15:37:20.000000000 -0300
+++ linux-2.6.38.2-gabix/drivers/video/logo/Kconfig 2011-05-06 \
        16:08:49.220118332 -0300
@@ -27,6 +27,10 @@
  bool "Standard 224-color Linux logo"
  default y

+config LOGO_LNCC_CLUT224
+ bool "Tux com o logo do LNCC em 224 cores"
+ default y
+
 config LOGO_BLACKFIN_VGA16
  bool "16-colour Blackfin Processor Linux logo"
  depends on BLACKFIN
diff -ruN linux-2.6.38.2.vanilla/drivers/video/logo/logo.c \
          linux-2.6.38.2-gabix/drivers/video/logo/logo.c
--- linux-2.6.38.2.vanilla/drivers/video/logo/logo.c 2011-03-27 \
          15:37:20.000000000 -0300
+++ linux-2.6.38.2-gabix/drivers/video/logo/logo.c 2011-05-06 \
          16:14:20.034104927 -0300
@@ -63,13 +63,17 @@
  }

  if (depth >= 8) {
+#ifdef CONFIG_LOGO_BLACKFIN_CLUT224
+ /* Blackfin Linux logo */
+ logo = &logo_blackfin_clut224;
+#endif
 #ifdef CONFIG_LOGO_LINUX_CLUT224
  /* Generic Linux logo */
  logo = &logo_linux_clut224;
 #endif
-#ifdef CONFIG_LOGO_BLACKFIN_CLUT224
-/* Blackfin Linux logo */
-logo = &logo_blackfin_clut224;
+#ifdef CONFIG_LOGO_LNCC_CLUT224
+ /* Tux com o logo do LNCC */
+ logo = &logo_lncc_clut224;
 #endif
 #ifdef CONFIG_LOGO_DEC_CLUT224
  /* DEC Linux logo on MIPS/MIPS64 or ALPHA */
diff -ruN linux-2.6.38.2.vanilla/drivers/video/logo/logo_lncc_clut224.ppm \
          linux-2.6.38.2-gabix/drivers/video/logo/logo_lncc_clut224.ppm
--- linux-2.6.38.2.vanilla/drivers/video/logo/logo_lncc_clut224.ppm \
          1969-12-31 21:00:00.000000000 -0300
```

```
+++ linux-2.6.38.2-gabix/drivers/video/logo/logo_lncc_clut224.ppm \
          2011-05-06 15:45:59.984107040 -0300
@@ -0,0 +1,1011 @@
+P3
+71 84
+255
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  3 6 1
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  45 46 44
+76 78 75  28 30 28  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  65 66 64
+126 127 125  85 87 84  45 46 44  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  69 71 68
+69 71 68  41 43 41  18 19 17  1 4 0  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  21 23 20
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
```

```
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  41 43 41  81 82 80  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+3 6 1  121 123 120  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  1 4 0
+106 108 105  52 53 51  1 4 0  57 59 56  1 4 0  1 4 0
+1 4 0  1 4 0  21 23 20  138 140 137  147 149 146  143 144 141
+121 123 120  1 4 0  65 66 64  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  171 173 170
+201 203 200  210 212 209  171 173 170  14 16 12  1 4 0  1 4 0
+1 4 0  1 4 0  201 203 200  232 234 231  232 234 231  232 234 231
+214 216 213  100 102 99  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  110 112 109  254 255 252
+254 255 252  254 255 252  254 255 252  1 4 0  1 4 0  1 4 0
+1 4 0  239 241 238  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  239 241 238  1 4 0
+1 4 0  1 4 0  254 255 252  249 251 248  1 4 0  1 4 0
+1 4 0  254 255 252  254 255 252  1 4 0  3 6 1  147 149 146
+254 255 252  254 255 252  197 199 196  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  239 241 238  1 4 0
+1 4 0  151 153 150  254 255 252  249 251 248  1 4 0  100 102 99
+14 16 12  254 255 252  254 255 252  1 4 0  1 4 0  143 144 141
+1 4 0  254 255 252  249 251 248  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  176 178 175  1 4 0
+1 4 0  147 149 146  254 255 252  186 188 185  1 4 0  1 4 0
+1 4 0  254 255 252  254 255 252  1 4 0  1 4 0  1 4 0
+1 4 0  254 255 252  245 247 243  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  222 224 221
+1 4 0  1 4 0  185 128 0  255 202 0  255 202 0  255 202 0
+248 196 0  255 202 0  181 125 1  1 4 0  1 4 0  1 4 0
+81 82 80  254 255 252  171 173 170  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  254 255 252
+21 23 20  181 125 1  189 132 0  255 202 0  255 202 0  254 208 32
+233 179 0  215 158 4  255 202 0  242 192 1  1 4 0  3 6 1
+254 255 252  254 255 252  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  49 50 48
+189 132 0  255 202 0  255 202 0  255 202 0  255 202 0  254 209 47
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  8 1 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  207 151 0
+255 202 0  255 202 0  255 202 0  255 202 0  254 209 47  255 202 0
+255 202 0  255 202 0  253 207 8  254 209 47  254 209 47  255 202 0
+255 202 0  255 202 0  160 98 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  170 111 0  239 189 0
+255 202 0  255 202 0  255 202 0  255 202 0  254 208 32  255 202 0
+255 202 0  255 202 0  254 209 47  254 209 47  255 202 0  255 202 0
+117 66 0  255 202 0  180 119 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  173 113 2  255 202 0
```

```
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  254 209 47  254 209 47  255 202 0  255 202 0  134 83 0
+255 202 0  255 202 0  33 19 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  3 6 1  164 101 1
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+254 208 32  255 202 0  255 202 0  107 57 0  255 202 0  255 202 0
+255 202 0  200 146 0  1 4 0  1 4 0  1 4 0  100 102 99
+100 102 99  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  160 98 0
+198 139 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  117 66 0  255 202 0  255 202 0  255 202 0  239 189 0
+189 132 0  163 165 162  100 102 99  1 4 0  1 4 0  91 93 90
+100 102 99  100 102 99  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  159 161 158
+126 127 125  198 139 0  251 199 0  251 199 0  255 202 0  255 202 0
+255 202 0  255 202 0  255 202 0  228 174 0  172 107 0  197 199 196
+210 212 209  210 212 209  171 173 170  1 4 0  1 4 0  1 4 0
+100 102 99  100 102 99  69 71 68  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  197 199 196
+167 169 166  121 123 120  189 132 0  241 185 0  251 199 0  251 199 0
+241 185 0  222 164 0  184 123 1  189 191 188  218 220 217  242 244 241
+254 255 252  254 255 252  239 241 238  1 4 0  1 4 0  1 4 0
+1 4 0  25 26 24  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  218 220 217
+206 208 205  155 156 154  116 117 114  180 119 0  189 126 0  189 126 0
+180 119 0  160 98 0  186 188 185  218 220 217  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  163 165 162  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  254 255 252
+254 255 252  192 194 191  155 156 154  134 136 133  138 140 137  147 149 146
+171 173 170  192 194 191  227 229 226  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  1 4 0  1 4 0  245 247 243  254 255 252
+254 255 252  236 238 235  189 191 188  167 169 166  171 173 170  186 188 185
+210 212 209  232 234 231  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  210 212 209  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  1 4 0  116 117 114  254 255 252  254 255 252
+254 255 252  254 255 252  227 229 226  206 208 205  206 208 205  222 224 221
+245 247 243  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  1 4 0  1 4 0  249 251 248  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  232 234 231  236 238 235  249 251 248
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
```

```
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  3 6 1
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+1 4 0  1 4 0  197 199 196  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  239 241 238
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+1 4 0  1 4 0  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+1 4 0  116 117 114  249 251 248  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  249 251 248  249 251 248  254 255 252
+163 165 162  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  121 123 120  189 191 188  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  197 199 196  254 255 252  236 238 235  218 220 217
+254 255 252  254 255 252  254 255 252  254 255 252  249 251 248  232 234 231
+206 208 205  189 191 188  179 181 178  171 173 170  171 173 170  239 241 238
+254 255 252  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  1 4 0
+1 4 0  126 127 125  167 169 166  197 199 196  218 220 217  8 1 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  254 255 252  254 255 252  254 255 252  249 251 248  222 224 221
+192 194 191  167 169 166  155 156 154  0 1 0  147 149 146  151 153 150
+239 241 238  49 50 48  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  1 4 0
+69 71 68  171 173 170  222 224 221  232 234 231  1 4 0  1 4 0
+254 255 252  254 255 252  254 255 252  254 255 252  190 187 191  1 4 0
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  245 247 243  222 224 221  0 1 0  147 149 146  151 153 150
+151 153 150  249 251 248  1 4 0  1 4 0  61 62 60  31 32 30
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  1 4 0  8 10 6  1 4 0  1 4 0
+167 169 166  254 255 252  254 255 252  8 1 0  254 255 252  1 4 0
+254 255 252  254 255 252  254 255 252  254 255 252  1 4 0  192 194 191
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  0 1 0  159 161 158  192 194 191
+151 153 150  189 191 188  1 4 0  1 4 0  1 4 0  3 6 1
+54 56 53  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  1 4 0  41 43 41  1 4 0  1 4 0
+254 255 252  254 255 252  1 4 0  21 23 20  37 36 39  8 1 0
+38 39 37  34 36 34  41 43 41  1 4 0  254 255 252  254 255 252
+8 1 0  8 1 0  3 6 1  3 6 1  8 1 0  8 1 0
+8 1 0  34 36 34  254 255 252  0 1 0  0 1 0  0 1 0
+0 1 0  155 156 154  254 255 252  1 4 0  1 4 0  1 4 0
+1 4 0  28 30 28  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  31 32 30  8 10 6  1 4 0  192 194 191
+254 255 252  147 149 146  0 1 0  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  0 1 0  179 181 178  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  214 216 213  206 208 205  14 16 12  1 4 0  18 19 17
```

```
+14 16 12  3 6 1  8 10 6  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  8 10 6  49 50 48  1 4 0  73 74 72  254 255 252
+254 255 252  147 149 146  0 1 0  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  0 1 0  186 188 185  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  0 1 0  8 10 6  254 255 252
+1 4 0  254 255 252  239 241 238  254 255 252  1 4 0  38 39 37
+38 39 37  14 16 12  52 53 51  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  41 43 41  8 10 6  1 4 0  254 255 252  254 255 252
+254 255 252  151 148 153  0 1 0  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  0 1 0  186 188 185  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  0 1 0  81 82 80  254 255 252
+3 6 1  254 255 252  254 255 252  254 255 252  1 4 0  25 26 24
+28 30 28  1 4 0  8 10 6  8 10 6  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+3 6 1  49 50 48  1 4 0  61 62 60  254 255 252  254 255 252
+254 255 252  1 4 0  0 1 0  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  0 1 0  186 188 185  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  0 1 0  212 210 214  21 23 20
+3 6 1  254 255 252  254 255 252  254 255 252  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  65 66 64  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+14 16 12  21 23 20  1 4 0  210 212 209  254 255 252  254 255 252
+1 4 0  167 165 169  0 1 0  254 255 252  254 255 252  254 255 252
+254 255 252  8 1 0  0 1 0  183 185 182  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  0 1 0  212 210 214  76 78 75
+3 6 1  254 255 252  254 255 252  254 255 252  116 117 114  1 4 0
+1 4 0  1 4 0  1 4 0  69 71 68  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+18 19 17  1 4 0  1 4 0  254 255 252  167 165 169  1 4 0
```

```
+254 255 252  151 148 153  0 1 0  254 255 252  254 255 252  254 255 252
+1 4 0  254 255 252  0 1 0  186 188 185  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  245 247 243  1 4 0
+1 4 0  1 4 0  1 4 0  54 56 53  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+3 6 1  1 4 0  1 4 0  254 255 252  3 6 1  3 6 1
+1 4 0  1 4 0  3 6 1  1 4 0  1 4 0  1 4 0
+206 208 205  254 255 252  0 1 0  179 181 178  254 255 252  254 255 252
+8 1 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  31 29 33  254 255 252  254 255 252  254 255 252  0 1 0
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  34 36 34  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  3 6 1
+45 46 44  1 4 0  81 82 80  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  208 206 210  254 255 252  1 4 0
+254 255 252  254 255 252  254 255 252  44 42 46  254 255 252  254 255 252
+1 4 0  254 255 252  254 255 252  167 165 169  0 1 0  163 165 162
+0 1 0  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  8 10 6  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  14 16 12
+54 56 53  1 4 0  210 212 209  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  189 191 188  3 6 1  254 255 252
+254 255 252  254 255 252  254 255 252  44 42 46  254 255 252  1 4 0
+254 255 252  254 255 252  254 255 252  0 1 0  210 212 209  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  31 32 30  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  21 23 20
+54 56 53  1 4 0  249 251 248  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  3 6 1  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  44 42 46  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  82 80 84  0 1 0  254 255 252
+0 1 0  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  28 30 28  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  14 16 12
+69 71 68  1 4 0  249 251 248  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  171 173 170  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  44 42 46  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  155 156 154  0 1 0
+220 218 222  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  1 4 0
+69 71 68  1 4 0  249 251 248  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  197 199 196  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  41 43 41  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  1 4 0  100 102 99  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  1 4 0  1 4 0
+31 32 30  8 10 6  245 247 243  254 255 252  1 4 0  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  254 255 252  0 1 0  201 203 200  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  44 42 46  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  85 87 84  0 1 0
+167 169 166  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  25 26 24  49 50 48  8 10 6  25 26 24  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  251 199 0  255 202 0  255 202 0
+1 4 0  54 56 53  245 247 243  254 255 252  0 1 0  0 1 0
+0 1 0  0 1 0  0 1 0  0 1 0  0 1 0  1 4 0
+254 255 252  254 255 252  0 1 0  116 117 114  110 108 112  110 112 109
+17 15 19  110 112 109  110 108 112  1 4 0  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  61 62 60  0 1 0  254 255 252
+0 1 0  254 255 252  254 255 252  254 255 252  249 251 248  1 4 0
+76 78 75  1 4 0  1 4 0  1 4 0  8 10 6  52 53 51
+91 93 90  1 4 0  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  1 4 0  76 78 75  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+254 255 252  3 6 1  197 199 196  254 255 252  254 255 252  254 255 252
```

```
+53 51 54  254 255 252  1 4 0  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  0 1 0  201 203 200  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  3 6 1  14 16 12
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  100 102 99  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  189 126 0  255 202 0  254 208 32  254 209 47  254 209 47
+255 202 0  255 202 0  1 4 0  1 4 0  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+8 10 6  206 208 205  254 255 252  254 255 252  254 255 252  254 255 252
+54 56 53  1 4 0  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  147 149 146  0 1 0  254 255 252
+0 1 0  255 202 0  254 209 47  254 209 47  254 209 47  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  31 32 30  107 70 0  255 202 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  255 202 0  253 207 8  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  255 202 0  1 4 0  1 4 0  242 244 241
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  11 8 13
+3 6 1  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+17 15 19  151 153 150  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  0 1 0
+254 255 252  255 202 0  254 208 32  254 209 47  254 209 47  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  253 207 8  254 209 47  255 202 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+193 135 2  255 202 0  254 208 32  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  255 202 0  242 192 1  1 4 0  1 4 0
+249 251 248  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  251 199 0  254 208 32  254 208 32  221 169 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  251 199 0  254 209 47  254 209 47  255 202 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  189 132 0
+255 202 0  255 202 0  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 208 32  255 202 0  255 202 0  1 4 0  1 4 0
+1 4 0  183 185 182  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+206 208 205  228 174 0  255 202 0  255 202 0  211 155 0  160 98 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+180 119 0  254 208 32  255 202 0  255 202 0  255 202 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  198 139 0  233 179 0  225 166 0  221 169 0  251 199 0
+255 202 0  253 207 8  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 208 32  255 202 0  255 202 0  1 4 0
+1 4 0  1 4 0  8 10 6  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  210 212 209
+163 165 162  215 158 4  255 202 0  255 202 0  228 174 0  189 132 0
+164 101 1  80 50 0  1 4 0  1 4 0  154 87 0  184 123 1
+239 189 0  255 202 0  255 202 0  255 202 0  255 202 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  164 101 1
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  254 208 32  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  255 202 0  255 202 0  248 196 0
+1 4 0  1 4 0  1 4 0  1 4 0  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  201 203 200
+151 153 150  203 143 0  255 202 0  255 202 0  242 192 1  215 158 4
+189 132 0  180 119 0  177 116 0  180 119 0  193 135 2  221 169 0
+255 202 0  254 209 47  254 209 47  254 209 47  254 208 32  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  239 189 0
+255 202 0  255 202 0  255 202 0  253 207 8  253 207 8  253 207 8
+254 208 32  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 208 32  255 202 0  255 202 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  249 251 248
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  210 212 209
+159 161 158  193 135 2  255 202 0  255 202 0  255 202 0  242 192 1
+221 169 0  218 161 0  218 161 0  222 164 0  241 185 0  255 202 0
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  253 207 8
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  221 169 0
+255 202 0  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  253 207 8  255 202 0
+255 202 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  218 220 217
+163 165 162  189 126 0  248 196 0  255 202 0  255 202 0  255 202 0
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 208 32  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  189 132 0
+255 202 0  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 208 32  255 202 0
+255 202 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  222 224 221
```

```
+147 149 146  184 123 1  241 185 0  255 202 0  255 202 0  255 202 0
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  254 208 32
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 208 32  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  172 107 0
+255 202 0  255 202 0  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 208 32
+255 202 0  255 202 0  1 4 0  1 4 0  1 4 0  1 4 0
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  189 191 188
+1 4 0  184 123 1  233 179 0  255 202 0  255 202 0  255 202 0
+253 207 8  253 207 8  255 202 0  253 207 8  253 207 8  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  255 202 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  160 98 0
+255 202 0  255 202 0  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+255 202 0  255 202 0  248 196 0  1 4 0  147 149 146  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  1 4 0
+1 4 0  185 128 0  233 179 0  255 202 0  255 202 0  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  255 202 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  160 98 0
+255 202 0  255 202 0  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 208 32  255 202 0  255 202 0  96 97 95  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  1 4 0  1 4 0
+3 6 1  189 132 0  241 185 0  255 202 0  253 207 8  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 208 32
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  173 113 2
+255 202 0  253 207 8  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 208 32  255 202 0  255 202 0  255 202 0  210 212 209  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  1 4 0  1 4 0  1 4 0
+139 81 0  198 139 0  239 189 0  255 202 0  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  251 199 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  255 202 0
+255 202 0  254 208 32  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
```

```
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  255 202 0  255 202 0  209 147 0  249 251 248
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+160 98 0  207 151 0  251 199 0  255 202 0  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  248 196 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  255 202 0
+255 202 0  253 207 8  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  255 202 0  255 202 0  233 179 0  3 6 1
+245 247 243  254 255 252  254 255 252  254 255 252  254 255 252  254 255 252
+254 255 252  254 255 252  254 255 252  254 255 252  222 224 221  3 6 1
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+172 107 0  211 155 0  255 202 0  255 202 0  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  255 202 0
+255 202 0  255 202 0  255 202 0  253 207 8  254 208 32  254 208 32
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  255 202 0  255 202 0  255 202 0  233 179 0  180 119 0
+1 4 0  1 4 0  76 78 75  151 153 150  159 161 158  138 140 137
+49 50 48  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+173 113 2  218 161 0  255 202 0  253 207 8  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 208 32  233 179 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  255 202 0
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  255 202 0  255 202 0  253 207 8  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 209 47
+254 209 47  253 207 8  255 202 0  255 202 0  228 174 0  184 123 1
+54 32 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  8 1 0
+180 119 0  218 161 0  255 202 0  255 202 0  254 208 32  254 209 47
+254 209 47  254 209 47  254 209 47  254 209 47  254 209 47  254 208 32
+254 208 32  253 207 8  255 202 0  222 164 0  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  4 1 7  4 1 7
+228 174 0  228 174 0  228 174 0  233 179 0  233 179 0  241 185 0
+239 189 0  251 199 0  255 202 0  255 202 0  255 202 0  255 202 0
+253 207 8  254 208 32  254 209 47  254 209 47  254 209 47  254 208 32
+255 202 0  255 202 0  255 202 0  248 196 0  221 169 0  181 125 1
+126 75 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  54 32 0
+180 119 0  215 158 4  239 189 0  255 202 0  255 202 0  254 208 32
+254 208 32  254 208 32  254 208 32  253 207 8  255 202 0  255 202 0
```

```
+251 199 0  215 158 4  170 111 0  4 1 7  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  4 1 7  4 1 7
+4 1 7  164 101 1  180 119 0  184 123 1  193 135 2  203 143 0
+207 151 0  215 158 4  221 169 0  233 179 0  239 189 0  255 202 0
+255 202 0  255 202 0  255 202 0  255 202 0  255 202 0  255 202 0
+255 202 0  255 202 0  242 192 1  228 174 0  211 155 0  180 119 0
+117 66 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+1 4 0  1 4 0  1 4 0  1 4 0  1 4 0  46 26 0
+177 116 0  207 151 0  228 174 0  241 185 0  255 202 0  255 202 0
+255 202 0  255 202 0  255 202 0  255 202 0  241 185 0  221 169 0
+193 135 2  126 75 0  1 4 0  4 1 7  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  148 88 0
+160 98 0  172 107 0  180 119 0  189 132 0  203 143 0  215 158 4
+228 174 0  239 189 0  251 199 0  255 202 0  255 202 0  255 202 0
+251 199 0  239 189 0  225 166 0  209 147 0  189 132 0  170 111 0
+3 6 1  1 4 0  1 4 0  1 4 0  4 1 7  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  1 4 0  1 4 0  1 4 0  1 4 0  1 4 0
+170 111 0  193 135 2  211 155 0  225 166 0  233 179 0  241 185 0
+239 189 0  239 189 0  228 174 0  215 158 4  193 135 2  173 113 2
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+1 4 0  4 1 7  1 4 0  4 1 7  160 92 0  170 111 0
+185 128 0  203 143 0  211 155 0  221 169 0  228 174 0  224 171 0
+221 169 0  218 161 0  203 143 0  189 126 0  172 107 0  25 16 0
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  4 1 7
+4 1 7  1 4 0  4 1 7  1 4 0  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  1 4 0  4 1 7  1 4 0
+160 98 0  180 119 0  193 135 2  203 143 0  211 155 0  215 158 4
+215 158 4  211 155 0  203 143 0  180 119 0  160 98 0  1 4 0
+4 1 7  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+1 4 0  4 1 7  1 4 0  4 1 7  4 1 7  1 4 0
+139 81 0  172 107 0  180 119 0  189 132 0  198 139 0  203 143 0
+203 143 0  198 139 0  184 123 1  172 107 0  80 50 0  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  1 4 0  4 1 7  1 4 0
+97 57 0  172 107 0  180 119 0  185 128 0  189 132 0  193 135 2
+189 132 0  189 126 0  180 119 0  160 98 0  1 4 0  4 1 7
+4 1 7  1 4 0  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  148 88 0  164 101 1  170 111 0
+172 107 0  172 107 0  160 98 0  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  117 66 0  160 98 0  164 101 1  172 107 0  172 107 0
+168 104 0  160 98 0  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  4 1 7  1 4 0
+1 4 0  4 1 7  1 4 0  4 1 7  4 1 7  1 4 0
+4 1 7  1 4 0  1 4 0  4 1 7  1 4 0  1 4 0
+4 1 7  1 4 0  4 1 7  4 1 7  1 4 0  4 1 7
+4 1 7  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  1 4 0  4 1 7  1 4 0
+1 4 0  4 1 7  1 4 0  1 4 0  4 1 7  1 4 0
+4 1 7  4 1 7  1 4 0  4 1 7  1 4 0  1 4 0
+4 1 7  1 4 0  4 1 7  4 1 7  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  1 4 0  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  4 1 7  4 1 7  1 4 0
+4 1 7  1 4 0  1 4 0  4 1 7  1 4 0  1 4 0
+4 1 7  1 4 0  4 1 7  4 1 7  1 4 0  4 1 7
+4 1 7  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  1 4 0  4 1 7  4 1 7  4 1 7  4 1 7
+1 4 0  4 1 7  4 1 7  1 4 0  4 1 7  1 4 0
+1 4 0  4 1 7  1 4 0  1 4 0  4 1 7  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  1 4 0  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  1 4 0
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  1 4 0  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
```

```
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
+4 1 7  4 1 7  4 1 7  4 1 7  4 1 7
diff -ruN linux-2.6.38.2.vanilla/drivers/video/logo/Makefile \
          linux-2.6.38.2-gabix/drivers/video/logo/Makefile
--- linux-2.6.38.2.vanilla/drivers/video/logo/Makefile 2011-03-27 \
          15:37:20.000000000 -0300
+++ linux-2.6.38.2-gabix/drivers/video/logo/Makefile 2011-05-06 \
          16:09:11.987109464 -0300
@@ -4,6 +4,7 @@
 obj-$(CONFIG_LOGO_LINUX_MONO) += logo_linux_mono.o
 obj-$(CONFIG_LOGO_LINUX_VGA16) += logo_linux_vga16.o
 obj-$(CONFIG_LOGO_LINUX_CLUT224) += logo_linux_clut224.o
+obj-$(CONFIG_LOGO_LNCC_CLUT224) += logo_lncc_clut224.o
 obj-$(CONFIG_LOGO_BLACKFIN_CLUT224) += logo_blackfin_clut224.o
 obj-$(CONFIG_LOGO_BLACKFIN_VGA16) += logo_blackfin_vga16.o
 obj-$(CONFIG_LOGO_DEC_CLUT224) += logo_dec_clut224.o
diff -ruN linux-2.6.38.2.vanilla/include/linux/linux_logo.h \
          linux-2.6.38.2-gabix/include/linux/linux_logo.h
--- linux-2.6.38.2.vanilla/include/linux/linux_logo.h 2011-03-27 \
        15:37:20.000000000 -0300
+++ linux-2.6.38.2-gabix/include/linux/linux_logo.h 2011-05-06 \
        15:46:30.826109454 -0300
@@ -35,6 +35,7 @@
 extern const struct linux_logo logo_linux_mono;
 extern const struct linux_logo logo_linux_vga16;
 extern const struct linux_logo logo_linux_clut224;
+extern const struct linux_logo logo_lncc_clut224;
 extern const struct linux_logo logo_blackfin_vga16;
 extern const struct linux_logo logo_blackfin_clut224;
 extern const struct linux_logo logo_dec_clut224;
```