O slackware é a distribuição linux mais antiga ainda em atividade. Tendo sido criada por Patrick Volkerding em 1993, a partir da SLS.

Em todos esses anos, a distro conquistou ardorosos utilizadores, principalmente graças à sua filosofia de simplicidade e estabilidade.

Um produto de extrema qualidade para usuários com esta mesma característica. E este zine é de slacker para slacker.

slackware zine

31 de Janeiro de 2005 – Edição #7

Editorial

Feliz aniversário para nós!!! É isso mesmo, em janeiro de 2004 saiu o primeiro número do slackwarezine e, desde então, pontualmente, em todo mês ímpar sai uma edição nova!!

Gostaríamos de agradecer a todos os nossos leitores, a todos que deram suas sugestões, a todos que baixaram uma cópia e a todos que fizeram propaganda da zine para os amigos e para os nem tão amigos.

Agradecemos também aos nossos colaboradores, pois sabemos que escrever artigos dá trabalho e consome tempo, e ficamos muito felizes em ver que muita gente gosta da idéia de compartilhar seus conhecimentos escrevendo artigos para a zine. E, normalmente, nossos colaboradores são reincidentes -;).

O nosso presente para esse aniversário é mais uma edição recheada de ótimos artigos, tratando desde a configuração de servidores de e-mail (do já conhecido "mago" do Postfix, Deives "thefallen" Michellis) até a instalação de aplicativos para o desktop (amule e gaim-vv, em artigos respectivamente do Clayton e do r00tsh3lL, que apesar do nome de 313373 parece ser uma boa pessoa). Passando por uma gama bem variada de outros assuntos...

Esperamos que todos vocês gostem dessa edição e nos acompanhem em mais um ano de slackwarezine, seja lendo, opinando ou colaborando ou, de preferência, fazendo todas essas coisas ao mesmo tempo!!

Boa Leitura

Piter PUNK

índice

slack users

Instalando o slackware em notebooks

Resolvi escrever este artigo por dois motivos: o primeiro está relacionado a algumas "adversidades" encontradas durante a instalação do **slackware** 10.0 em um notebook Toshiba modelo A40-S161. Coloquei o termo adversidades entre aspas porque não ocorreu nada de mais durante a instalação, apenas algumas configurações mais específicas e menos comuns deram um pouco de trabalho, mas nada que não possa ser contornado;

O segundo motivo é acabar com o mito de que o linux, em especial o slack, não funciona com notebooks de "grife". Vamos ao trabalho!!!

Bem, o notebook em questão é todo baseado em tecnologia Intel, com exceção da leitora de cartões, de fabricação da própria Toshiba. Não irei detalhar a instalacão em si, mas apenas os pontos em que eu encontrei dificuldades. Desse modo, instale o seu slack da maneira que melhor lhe convir.

A instalação do S.O. ocorre sem problemas, a detecção de hardware também é bastante tranqüila e funcional, no entanto, alguns "ajustes", tornam-se necessários. O `alsaconf`, por exemplo, detecta normalmente a controladora de som. Detecta também a controladora de som que equipa o modem, como podemos ver na saída do comando `lspci` a seguir (`lspci -vv` para maiores detalhes):

```
00:1f.5 Multimedia audio controller: \
      Intel Corp. 82801DB (ICH4) \
      AC'97 Audio Controller (rev 03)
00:1f.6 Modem: Intel Corp. 82801DB (ICH4)\
      AC'97 Modem Controller (rev 03)
```

Bem, aí é que está um dos grandes problemas que enfrentei durante a configuração do sistema. O `hotplug` detecta normalmente os módulos ALSA da placa de som e os módulos de compatibilidade baseados no "antigo" sistema OSS. Além disso, são carregados também os módulos referentes a controladora wave do modem, que é o "grande" problema que enfrentaremos agora.

Bem, como todo (bom) usuário de linux, lemos as mensagens de boot e vimos que é necessários executar o `alsamixer`, setar os volumes desejados em nosso mixer e em seguida salvar as configurações, com o comando `alsactl store`.

Para nossa surpresa, o comando `alsamixer` não funciona, dizendo que não existe um dispositivo válido, e agora, o que fazer?

Pela minha interpretação (não necessariamente correta), esse problema ocorre porque o sistema detecta mais de uma controladora de som (controladora de som e a do modem) e coloca a controladora de som do modem como primária e, como não está corretamente configurada, talvez pelo fato de se tratar de um winmodem, causa esse problema.

Se utilizarmos o comando `alsamixer -C1`, as opções referentes a placa de som serão carregadas normalmente, no entanto, ao menos no gnome, o som não funcionava, mesmo com os canais de som corretamente configurados. Como não utilizo o modem nem no trabalho, nem em casa, optei pela saída mais prática que consiste no não carregamento dos módulos referente a ele. Para tanto, basta identificarmos o módulo referente ao wave device do modem, que no nosso caso é o módulo `snd-intel8x0m`.

Com o nome do módulo em mãos, vamos incluí-lo no final da `blacklist` do `hotplug` (`/etc/hotplug/blacklist`), desse modo, quando o dispositivo for detectado, o módulo referente a ele não será levantado, o que irá normalizar o funcionamento da placa de som do notebook. Vale lembrar que essa solução, apesar de prática, não é a mais indicada, uma vez que você estará provavelmente inutilizando um recurso do notebook, no caso, o modem. Acredito que uma outra saída é tentar definir um alias para os dispositivos, invertendo entre os primários e secundários, ou algo parecido.

Uma boa lida na documentação do ALSA também é bem vinda, existem instruções que dizem ao ALSA que ele deve se limitar a detecção de um único device (o que geraria o mesmo problema dessa solução, a não utilização de um recurso do modem), talvez existam parâmetros que permitam definir que o módulo "x" corresponda ao device primário e o "y" corresponda ao secundário, o que resolveria o problema facilmente. Eu, sinceramente, não procurei nada sobre isso. Se você utilizar o modem, já sabe qual o problema e como (provavelmente) contorná-lo.

continua...

Um outro problema está relacionado com o monitoramento da bateria. O slack não instala o kernel default (`bare.i`) com suporte a ACPI nem a APM (não suportado por esse modelo de notebook), desse modo, uma das soluções seria instalar o slack com o kernel `bare_acpi.i`, mas acredito que a solução mais indicada é mesmo a recompilação do kernel, permitindo assim, o suporte a ACPI e a realiação de alguns "ajustes finos" essenciais para que seu **slackware** renda 100%, tais como a otimização para cpu's P4 e a remoção de módulos desnecessários não utilizados pelo seu sistema/equipamento.

"Caminho" dos módulos dentro do menu de configuração do kernel:

```
ACPI: General Setup, ACPI Support
      [*] ACPI Support
            <*>   AC Adapter
            <*>   Battery
            <*>   Button
            <*>   Fan
            <*>   Processor
            <*>   Thermal Zone
            <*>   Toshiba Laptop Extras
```

Habilitar também: `Processor type and features,<*> Toshiba Laptop support`

Após a compilação do kernel, com suporte aos módulos extra da Toshiba, o monitor de bateria funcionará corretamente, desde que o daemon `acpid` seja carregado na inicialização do sistema.

Para colocar o monitor de bateria no painel de seu gnome, clique com o botão direito sobre ele, ao lado do relógio, por exemplo. Clique em "`Add to Panel, Utility, Battery Charge Monitor`".

As funcionalidades adicionadas pelos módulos-extra da Toshiba, podem ser visualizadas em:

```
/proc/acpi/toshiba
```

Vale lembrar que, para ajustar a intensidade de brilho do LCD, é necessário ajustar manualmente esse parâmetro, como a seguir:

```
# cat /proc/acpi/toshiba/lcd
brightness:              7
brightness_levels:       8
```

Para alterar, basta editar esse arquivo. Uma boa hora para treinar seus conhecimentos em shell e desenvolver um utilitário para isso... :) Além disso, é possível verificar a temperatura da CPU, o estado da bateria, entre muitas outras opções, tudo isso em `/proc/acpi`.

Por fim, devemos habilitar a aceleradora 3D intel 830, que não é configurada de imediato. No meu caso, a controladora de vídeo utilizada pelo `X.org` foi a padrão VESA, que apesar de trabalhar na resolução máxima suportada pelo dispositivo, não oferece aceleração 3D.

Para habilitar a configuração 3D, basta executar (em modo texto) o `xorgsetup`. ele irá detectar a controladora i810 e o módulo de aceleração 3D baseado em i830, inicie o modo gráfico e pronto!!!

Em alguns casos, ao testar o funcionamento da aceleração 3D (o aplicativo `glxgears` é uma boa pedida) seu slack irá reclamar sobre as permissões referentes a operação em modo OpenGL, para corrigir o problema, adicione (como root) as seguintes linhas no seu `xorg.conf` (em `/etc/X11/`):

```
Section "DRI"
     Mode 0666
EndSection
```

Em seguida reinicie seu ambiente gráfico e teste novamente o `glxgears`, tudo deve funcionar.

Uma última dica fica por conta da gravadora de CD's. Não se esqueça de habilitar a emulação SCSI, habilitando o módulo `/sbin/modprobe ide-scsi` em `/etc/rc.d/rc.modules` e adicionado a linha:

```
append="hdc=ide-scsi"
```

em seu `lilo.conf`, substituindo "`hdc`" pelo dispositivo referente a seu gravador de CD's. Em seguida, salve o arquivo e o atualize digitando "`lilo`", dê um boot na máquina e utilize seu `cdrecord` a vontade (ou `xcdroast` pra quem gosta de uma interface gráfica).

Bem, espero que esse tutorial seja útil para quem precisa instalar o slack nesse note ou em modelos de configuração similar.

Grande abraço a todos,

Clayton Eduardo dos Santos
<claytones@terra.com.br>

continua...

Adicionando verificação de SPF ao Postfix

Introdução

Talvez você já tenha ouvido falar "desse tal de SPF" e se pergunte que novidade é essa que apareceu na Internet. A sigla significa "Sender Policy Framework", ou "Estrutura de Politicas de Remetente". Diz a lenda que originalmente a sigla significava "Sender Permitted From", mas que foi alterada para refletir melhor o tipo de trabalho e ferramentas que a idéia do SPF traria.

Nesse artigo vamos ver pra que serve o SPF, como ele ajuda o controle de email mundial e como usá-lo, tanto do lado "cliente" como "servidor".

1. Para que serve

O SPF basicamente serve para dizer aos servidores de email espalhados pela Internet quais endereços IP/servidores estão autorizados a enviar email (rementente) com o domínio designado.

Assim, eu, como administrador do dominio `xyz.com.br`, publicarei um registro SPF dizendo que os servidores `200.200.200.200` e `200.200.200.201` estão autorizados a enviar emails com remetentes `@xyz.com.br`; quaisquer outros servidores tentando enviar emails como `@xyz.com.br` são falsos, e tais emails devem ser rejeitados.

De maneira similar, quando eu estiver recebendo um email de um remetente `fulano@qwerty.com.br`, verificarei seu registro SPF para saber se o servidor tentando me passar esse email tem autorização para isso.

Note que o SPF não se trata necessariamente de uma ferramenta anti-SPAM. Embora uma quantidade razoável de SPAM possa ser combatido com o SPF, estatísticas mostram que a maior parte dos spammers "profissionais" já tinham registros SPF implantados, ao passo que apenas uma pequena parcela dos dominios de emails legítimos tinham o registro.

Nesse contexto, o SPF ajuda a combater outra forma de lixo "internético": os Scams ou Phishing Scams, pessoas enviando emails como se fossem de bancos ou similares pedindo para que você acesse uma página e "atualize seus dados".

Você talvez esteja pensando: "Mas se esse negocio não é a arma final contra o SPAM vou implementar pra que?". O SPF ajuda a impedir emails falsificados, dizendo ser você. Isso ajuda a prevenir fraudes e similares. Também dificulta a ação de alguns vírus, que sorteiam um nome na lista de contatos da vítima para usar como remetente.

2. Como funciona

Registros SPF são simples registros texto na tabela de DNS de seu domínio. Esses registros possuem sintaxe própria, que pode ser consultada direto no site oficial do SPF - `http://spf.pobox.com/` . Há inclusive um "Wizard" para gerar o registro SPF para você, e é recomendado que você use esse wizard quando for implementar o registro em seu domínio.

Se você usa o BIND como DNS, o registro ficaria mais ou menos assim na zona de DNS:

```
.     IN  TXT "v=spf1 mx -all"
```

Se quiser saber qual o registro SPF de um domínio, basta rodar essa consulta:

```
thefallen@KlingonRealm:~$ host -t txt \
      dominio.com.br
dominio.com.br text "v=spf1 mx -all"
```

A sintaxe mencionada acima diz que apenas os servidores listados como MX do domínio estão autorizados a mandar email em nome do domínio, e que o registro é a "autoridade final" (`-all`), quer dizer, pode-se rejeitar a mensagem se não sair de algum MX do dominio. Se a palavra-chave fosse "`?all`" quer dizer que o registro é "neutro", ou está em período de implantação, e ainda podem haver servidores fora dessa lista; portanto NÃO se deve rejeitar a mensagem. Uma lidinha na especificação do site do SPF ajuda a esclarecer esse ponto.

3. Quando a verificação de SPF funciona

Uma dúvida que você talvez tenha é "Se eu ativar SPF, vou parar de receber email de quem não tem SPF?". A verificação de SPF só acontece quando o domínio do remente já publicou o registro SPF. Assim, se o site do remetente não tiver aderido ao SPF, os emails vão passar normalmente (poderão ser forjados, visto que não há o registro SPF).

continua...

continua...

Se você não quiser ativar a verificação do SPF em seu MTA agora, pode fazer apenas o primeiro passo da implantação, que é adicionar o registro no DNS. Isso também não vai comprometer emails chegando ou saindo de/para sites que não façam a verificação SPF.

4. Implementações de verificação SPF

Existem varias implementações de verificação de SPF. Além da implementação inicial em Perl (`Mail::SPF::Query`), há bibliotecas especializadas (`libspf` e `libspf2`), módulos em Python, módulos para SpamAssassin, módulos pro Milter do Sendmail, patches para verificação nativa em Postfix, Exim e Qmail, e inclusive está listado no site uma implementação para Exchange.

No Howto a seguir usaremos a implementação via Policy Daemon do Postfix (a `policyd` listada no site). Não precisa de patches no Postfix e o daemon adicional é bastante leve para permitir que o sistema escale bem para ambientes maiores.

5. Para os mais apressados

Para os que não quiseram ler a teoria acima, a implementação do SPF ocorre em 2 partes:

- Publicar um registro SPF em seu DNS
- Implementar a verificação do registro SPF no MTA

5.1. Gerando o registro SPF

Va até o site do Wizard do SPF (`http://spf.pobox.com/wizard.html`) e responda as perguntas. Na maioria dos casos, o registro fica como "`v=spf1 mx -all`"

Edite a zona DNS de seu dominio no BIND/named e adicione a seguinte linha no final do arquivo:

```
.     IN TXT "v=spf1 mx -all"
```

5.2. Implementando a verificação no MTA

Primeiro de tudo, tenha em mente que o Policy Daemon Delegation (check_policy_service) só está disponível a partir da versão 2.1 do Postfix. Para saber qual versão você esta rodando, digite o comando:

```
root@KlingonRealm:~# /usr/sbin/postconf \
                        mail_version
mail_version = 2.1.5
```

Para implantar o `policyd` para verificar os registros de servidores conectando ao seu MTA, vai precisar da `libspf2` e do `policyd` (`http://www.libspf2.org`).

Note que usaremos a versão do site `www.libspf2.org`, que é a versão em C que usa a `libspf2`. O `policyd` disponível no site `http://spf.pobox.com/` é a versão em Perl, consideravelmente mais pesada e não tão bem escalável.

A instalação da biblioteca `libspf2` é bastante simples:

```
# ./configure --prefix=/usr
# make
# su -c "make install"
```

Se quiser já instalar o `policyd` direto no `/usr/libexec/postfix`, rode o seguinte comando:

```
# ./configure --prefix=/usr \
      --sbindir=/usr/libexec/postfix
# make
# su -c "make install"
```

Precisamos agora referenciar o serviço no `master.cf`:

```
# /etc/postfix/master.cf:
policy    unix  -  n  n  -  -  spawn
      user=nobody \
      argv=/usr/libexec/postfix/policyd
```

Basta adicionar a seguinte linha no `main.cf` para que ele já faça a verificação:

```
# /etc/postfix/main.cf:
smtpd_sender_restrictions = \
            suas_restricoes_vao_aqui,
          permit_mynetworks,
          check_policy_service \
            unix:private/policy
```

Note que é necessário o `permit_mynetworks` para que o SPF não tente verificar o próprio domínio. Se quiser mover a verificação de SPF pra "frente" das regras, não se esqueca de mover também o `permit_mynetworks`. Tente agora enviar um email de algum IP externo com a seguinte sintaxe:

```
thefallen@KlingonRealm:~$ telnet \
      mail.seudominio.com.br 25
220 mail.seudominio.com.br SMTP Postfix
MAIL FROM: <seuemail@seudominio.com.br>
250 Ok
RCPT TO: <seuemail@seudominio.com.br>
554 <seuemail@seudominio.com.br>: Sender \
address rejected: Please see \
http://spf.pobox.com/why.html?\
sender=seuemail40seudominio.com.br&\
ip=201.x.y.x&receiver=\
mail.seudominio.com.br
QUIT
```

continua...

continua...

6. O que fazer quandos as coisas não dão certo

Primeiro, verifique o log de email (normalmente `/var/log/maillog`). Ele é seu melhor amigo nessa hora :)

Se não houver nenhuma mensagem de erro lá, verifique as configurações que vc acabou de fazer, e certifique-se que rodou o comando "`postfix reload`" ou "`postfix stop; postfix start`".

Verifique se o SPF foi corretamente instalado com o comando `spfquery`:

```
thefallen@Ragnarok:~$ /usr/bin/spfquery \
      -ip 200.200.200.200 \
      -sender fulano@uol.com.br \
      -helo fulano
  fail
  Please see \
      http://spf.pobox.com/why.html?\
      sender=fulano%40uol.com.br&\
      ip=200.200.200.200&\
      receiver=spfquery
  spfquery: domain of uol.com.br does \
      not designate 200.200.200.200 as \
      permitted sender
  Received-SPF: fail (spfquery: domain \
      of uol.com.br does not designate \
      200.200.200.200 as permitted \
      sender) \
      client-ip=200.200.200.200; \
      envelope-from=fulano@uol.com.br; \
      helo=fulano;
```

Se todo o resto não deu certo, você pode tentar (na seguinte ordem :D) pesquisar no Google, as listas de discussão do Postfix (`http://www.postfix.org/lists.html`) e, em último caso, canais de IRC (os que freqüento são `#postfix` e `#postfix-br` em `irc.freenode.net`).

Deives “thefallen” Michellis
<dmichell@grupogeo.com.br>

Rodando Aplicações Remotas Seguras

Neste artigo vou demonstrar como rodar programas gráficos de outro computador utilizando uma conexão segura criptografada, usando o SSH, sem necessitar toda uma sessão de X aberta remotamente.

A configuração é bem simples, se resume em descomentar uma linha do arquivo de configuração do SSH e reiniciá-lo. Então vamos lá! No computador que irá servir de servidor, edite o arquivo `/etc/ssh/sshd_config` descomentando a linha que contêm `X11Forwarding` e troque nesta linha onde está 'no' por 'yes':

Ela habilita o forwarding e seta da variável de ambiente `DISPLAY` para o hostname do cliente. Você pode ter vários clientes conectados a esse servidor e cada um rodando uma aplicação diferente.

Agora, reinicie seu `sshd`, como root:

```
root@servidor:~# /etc/rc.d/rc.sshd \
                  restart
```

No computador 'burro', que irá apenas abrir os programas do servidor, conecte usando a flag `-X` do ssh, assim:

```
toledo@local:~$ ssh -X IP
```

A flag `-X`, como o próprio nome diz, interpreta que estamos aberto a receber forwarding do X. Essa opção pode ser usada em qualquer sessão SSH, mas se o servidor não estiver habilitado ao `X11Forwarding`, não funcionará.

Após o logon execute o programa que desejar, por exemplo:

```
toledo@servidor:~$ mozilla
```

Em instantes o programa abrirá em seu computador (não esqueça de estar dentro uma sessão do X, já que os programas são gráficos ;))

Abrindo apenas os programas criptografados você consome menos recursos e memória do servidor do que abrir o X inteiro. E, o impacto da criptografia é mínimo na performance e máximo na segurança.

toledo <toledo@core-dumped.org>

Instalando o amule no slack

Nesse artigo, fiz um passo a passo, mostrando como se instalar o amule no Linux.Pra quem não conhece, o amule e um clone free do emule, cliente p2p que utiliza a rede do edonkey para compartilhamento de arquivos na rede.

Vamos ao trabalho:

1o passo, baixar os arquivos necessários:

Baixe o conteúdo das seguintes URLs:

```
http://download.berlios.de/amule/\
      aMule-2.0.0rc8.tar.gz
http://download.berlios.de/amule/\
      crypto-5.2.1.tar.bz2
http://www.boutell.com/gd/http/\
      gd-2.0.26.tar.gz
http://download.berlios.de/amule/\
      wxBase-2.4.2.tar.gz
http://download.berlios.de/amule/\
      wxGTK-2.4.2.tar.gz
```

2o passo, instalando as dependências:

Instalando o wxGTK:

```
# tar -xvzf wxGTK-2.4.2.tar.gz
# cd wxGTK-2.4.2
# ./configure --prefix=/usr \
      --disable-gtk2 \
      --with-gtk
# make
# checkinstall
# ldconfig
```

Instalando o wxBase:

```
# tar -xvzf wxBase-2.4.2.tar.gz
# cd wxBase-2.4.2
# ./configure --prefix=/usr
# make
# checkinstall
# ln -sf /usr/bin/wxgtk-2.4-config \
      /usr/bin/wx-config
# ln -sf /usr/bin/wxbase-2.4-config \
      /usr/bin/wxbase-config
# ldconfig
```

Instalando a libcrypto:

```
# tar -xvjf crypto-5.2.1.tar.bz2
# cd crypto-5.2.1
# make
# cp libcryptopp.a /usr/lib/
# mkdir /usr/include/cryptopp
# cp *.h /usr/include/cryptopp/
# ldconfig
```

Instalando o pacote curl:

Se você ainda não o tem instalado, baixe de:

```
ftp://ftp.slackware.com/pub/slackware/\
      slackware-current/slackware/n/\
      curl-7.12.0-i486-1.tgz
```

E instale-o com o comando:

```
# installpkg curl-7.12.0-i486-1.tgz
```

3o passo, instalando o amule:

```
# tar -xvjf aMule-2.0.0rc5.tar.bz2
# cd aMule-2.0.0rc5
# ./configure --disable-debug \
      --enable-optimise
# make
# checkinstall
```

Finalmente, vamos executar o amule dentro do X, digitando:

```
# amule
```

Vale lembrar que existe um botão para fazer um update dos servidores dentro do próprio amule, no entanto, em alguns casos e possível que isso não funcione, uma vez que a url indicada pode estar momentaneamente fora do ar ou coisa do tipo, para solucionar esse problema, basta inserir os servidores manualmente. Uma boa lista de servidores de emule está disponível em:

```
http://emule-serverlist.gotdns.com/\
      serverlist/server.php
```

Até o próximo artigo,

Clayton Eduardo dos Santos
<claytones@terra.com.br>

Configurando o Slackintosh

No artigo passado, tratamos da instalação do **slackintosh** 8.1, e dissemos que a configuração era assunto para outro artigo. Bom, este é o tal "outro artigo". Com ele, iremos ter um sistema totalmente funcional, com som, rede, teclado, mouse e X todos configurados e prontinhos para usar.

Claro, que a primeira coisa a fazer é sumir com aquele bizonho prompt do OpenFirmware (assim, você não vai ver mais o prompt do OpenFirmware, a não ser que pressione Option+Command+O+F)

```
# nvsetenv auto-boot? true
```

A configuração de softwares como sendmail, mozilla e apache (entre muitos outros) é idêntica a configuração que faríamos em um PC, por esse motivo, este artigo aborda apenas configurações particulares do Macintosh (que se concentram no hardware).

Para várias das configurações aqui indicadas, é necessário recompilar o kernel. O código fonte do mesmo se encontra na série K, enquanto os pacotes necessários para a compilação estão na série D. Vamos ao ataque...

Teclado e Mouse

Não sei se você já percebeu no seu Macintosh, mas o mouse que vem com ele possui apenas um botão. E, se você usa o X, já deve ter percebido como isso não é legal... A solução mais óbvia é simplesmente comprar uma placa PCI/USB e colocar um mouse USB com váááários botões e, de preferência, com a famosa rodinha... -:).

Infelizmente esta solução para mim foi totalmente infuncional, graças a um pequeno problema: o 5500 possui apenas um slot PCI e nele está a minha placa de rede. Entre rede e mouse de três botões, fico com a rede.

Continuando a liste de probemas e peculiaridades, por default, o teclado do Macintosh manda keycodes de Macintosh. No console a gente não vê muita diferença, mas programas que acessam diretamente o teclado (alguém mencionou X?) possuem muitos problemas com isso e, se temos um Macintosh, queremos o modo gráfico!! Afinal, mac só com texto chega a ser uma heresia...

Com os problemas já expostos, podemos agora pensar na solução. Aliás, nem precisamos pensar muito nisso pois os sábios desenvolvedores do kernel já pensaram antes pela gente. Para tudo correr perfeitamente, basta que ativemos alguns parâmetros no kernel, com o comando sysctl.

Primeiro vamos cuidar dos keycodes, padronizando-os para ao invés de mandar códigos de Mac, mandar os convencionais do Linux:

```
# sysctl -w \
      dev/mac_hid/\
      keyboard_sends_linux_keycodes=1
```

Com isso solucionado, podemos agora setar algumas das teclas para funcionarem como botões dois e três (respectivamente, botão do meio e da direita). Antes de mais nada, vamos ativar esse recurso:

```
# sysctl -w \
      dev/mac_hid/\
            mouse_button_emulation=1
```

Particularmente eu gostei de deixar o botão da direita como sendo a tecla Command e o botão do meio mapeado para o Control da direita (que nunca uso):

```
# sysctl -w \
      dev/mac_hid/\
      mouse_button3_keycode=125
# sysctl -w \
      dev/mac_hid/\
      mouse_button2_keycode=97
```

Os números 125 e 97 correspondem às teclas que mencionei. Se você quiser colocar os botões em outras teclas, use o comando showkey (tem que
estar no console). Anote os keycodes das teclas que escolher (muita gente gosta de usar o F11 e F12 para isso) e espere 10 segundos (sem apertar nenhuma tecla) para sair automagicamente do showkey.

Ah! E para o mouse funcionar direitinho no console, rode o mouseconfig e selecione o seu mouse como se fosse USB. Se quiser fazer isso manualmente basta fazer um link do /dev/input/mice para o /dev/mouse.

continua...

continua...

Som

Essa configuração é uma das mais simples! -:)

```
# modprobe dmasound_pmac
```

Pronto! Foi carregado o módulo de som para o seu PowerMac! Não sei se são todos os PowerMacs, mas a grande maioria usa o chip de som AWACS, e esse é o módulo para ele. Para regular o volume, use o `rexima`.

Rede

A configuração de rede é outra que não traz maiores problemas. Para quem está acostumado com o `netconfig` do **`slackware`**, basta utilizá-lo. Depois de configurar a rede, é só rodar o `rc.inet1`:

```
# /etc/rc.d/rc.inet1
```

E tudo estará OK. Ou quase tudo, no meu caso foi detectado o módulo da placa de rede porque uso uma placa PCI supercomum (Realtek 8139). Provavelmente, se você tiver outra placa, talvez precise carregar o módulo manualmente e, só depois rodar o `netconfig`. Por exemplo:

```
# modprobe mace
```

Carrega o módulo de rede "padrão" para Macintoshs. Se o seu Mac possui rede on-board, a chance de ser uma MACE, se não é de 100% é de uns bons 90%. Se o seu PowerMac é daqueles que possuem duas portas, uma AAUI e uma RJ-45, pode utilizar a opção `port_aaui` para forçar ou não a utilização da AAUI (use `port_aaui=1` ou `port_aaui=0` para isso).

Seriais

Os Macs não possuem porta paralela. Nem os OldWorld (que são o foco do nosso artigo) nem os NewWorld; nos primeiros, as impressoras são conectadas via porta serial e nos segundos via USB.

Ou seja, se você possui uma impressora para o seu PowerMac, vai precisar habilitar as portas seriais dele. Isso é feito carregando o módulo apropriado:

```
# modprobe macserial
```

Se você é novo no mundo do Macintosh, as entradas seriais são os conectores redondos com 9 pinos na parte traseira dele. O conector redondo com 4 pinos é ADB e o que parece uma porta paralela é SCSI.

Vídeo

Vamos ter que começar com um pouco de história. Uma das principais características do Macintosh, desde o seu lançamento, é a ausência de um "modo texto". Ou seja, ele apenas possui o modo gráfico. O que era um problema razoável para rodar o Linux nele.

Para resolver esse problema, foi criado o framebuffer que nada mais é do que mapear a tela gráfica como se fosse um terminal de texto, ou seja, escrevemos texto diretamente no modo gráfico. E é assim que trabalha o Macintosh, tendo sido criado, inclusive, um X que funcione sobre o framebuffer. Depois tanto o framebuffer como o X que roda sobre ele foram portados para o x86. Mas isso não faz parte desse artigo.

O que faz parte desse artigo é que o jeito mais fácil (e algumas vezes o único) de ter o X rodando em um Macintosh é rodando-o sobre o framebuffer. Apesar do comando fbset poder alterar a resolução do framebuffer on-the-fly, eu nunca fui muito feliz com ele e, aliás, não fico trocando de resolução toda hora. Nesse caso, o melhor a fazer é configurar o seu `quik.conf` e colocar nele a resolução adequada. Alguns detalhes antes de começar:

- Quando no modo de 24bits, o RGB do Mac na realidade é BGR e, no X, todas as cores ficam trocadas. Não sei se o X em versões mais novas ainda possui esse problema, mas o que vem no Slackintosh 8.1 com certeza padece desse mal. Ah! 24bits e 32bits são praticamente a mesma coisa.

- No modo de 16 bits, ao invés de utilizar 5-6-5 (5 bits para R, 6 para G e 5 para B) que é praticamente um padrão no PC, são utilizados 5 bits para cada um dos canais de cor. Traduzindo, 16 bits na realidade são 15.

- Freqüências não suportadas podem queimar o seu monitor!!! Tenha certeza antes de tentar algo, consulte o manual e toda a documentação que encontrar. 832x624x8bpp é uma configuração bem popular e funciona praticamente em qualquer PowerMac.

Agora que você já é uma pessoa bem informada, vamos ao que interessa: os modos de vídeo que você tem disponíveis.

Dê uma olhadinha e, por favor, NÃO me perguntem qual a diferença entre o modo `mac16` e o `mac17`. Essas coisas estão além da minha compreensão e, se alguém se preocupar em procurar o porquê, por favor me avise -;)

continua...

continua...

Depois de decidir qual a resolução apropriada, você precisa também decidir qual a profundidade. 8bits de cor lhe dão 256 cores, 16bits que na realidade são 15 lhe dão 32768 cores e 32bits que na realidade são 24 lhe dão 16 milhões de cores.

Lembre-se que quanto mais bits de cor você usar, mais memória irá gastar. Memória de vídeo que, normalmente, não é expansível. Para saber quanto de memória você irá precisar em bytes basta fazer a seguinte conta:

```
resx * resy * bits/8
```

Por exemplo, para 832x624 com 16bits de cor você irá gastar:

```
832*624*16/8 = 1038336 bytes =
             = 1014 kbytes = 0.99 Mbyte
```

Quando for usar os modos de 15 e 24 bits faça as contas como sendo 16 e 32 bits, respectivamente. Agora que já sabemos a resolução e a profundidade que queremos, chegou a hora de passarmos essa informação para o kernel, através do `quik`.

Isso é particularmente simples. Por exemplo eu utilizo 1024x768@60 com 16bits de cor. Olhando na **Tabela 1**, vemos que a resolução 1024x768 a 60Hz é o modo `mac14`. Assim, eu adiciono a seguinte linha:

```
append="video=atyfb:mac14-16"
```

Coloque-a logo abaixo da linha que diz

```
image=/vmlinux
```

Em detalhes, o que essa linha está dizendo é: "Como vídeo, use o frambuffer da ATI, entre no modo mac14 com 16bits de cor"

O PowerMac 5500 usa como vídeo uma ATI 3D Rage I/II como é possível ver com a saída do `lspci`. A saída do lspci é útil em praticamente todos os Macintoshs PCI. Para quem gostaria de saber, disponibilizamos na **Tabela 2** uma lista dos dispositivos de framebuffer e os Macintoshs que usam cada um deles.

Se você não sabe qual está usando, escolha macfb que é o driver "padrão", mas o ideal é descobrir qual o vídeo que o seu Mac está usando. Antes de passarmos para o próximo item, é bom saber que existe um segundo método de escrever a linha do "`append`" que colocamos no `quik.conf`:

```
append="video=atyfb:vmode:14,cmode:16"
```

Alguns dos drivers só se dão bem com esse tipo de passagem de parâmetro, e esse pode ser o seu caso. Depois de tudo OK, reinstale o `quik`:

```
# quik -v -C /boot/quik.conf
```

No próximo boot deve aparecer um lindo framebuffer na resolução que você achou melhor.

Configurando o X

Com o framebuffer e o mouse configurados, falta muito pouco para ter um X totalmente funcional. É bem simples copiar o `/etc/X11/XF86Config-fbdev` para o `/etc/X11/XF86Config` e editar algumas coisinhas...

A configuração de teclado padrão US-acentos é:

```
Option "XkbRules"  "xfree86"
Option "XkbModel"  "pc102"
Option "XkbLayout" "us_intl"
```

Esse tipo de configuração é igual a configuração normal para x86, e costuma ter bastante informação a respeito pela rede. Se o seu teclado não for um US internacional com 102 teclas, basta colocar as opções apropriadas. Por exemplo, o ABNT2 usa "`abnt2`" como `XkbModel` e "`br`" como `XkbLayout`.

O protocolo do mouse ADB é PS/2 e o device `/dev/mouse` se você fez o link como falamos na parte deste artigo que trata da configuração do mouse. Isso é passado para o X pelas opções:

```
Option "Protocol" "PS/2"
Option "Device"   "/dev/mouse"
```

Dentro da seção "`InputDevice`", e dentro da sub-seção que trata da configuração de Mouse. Se te anima, isso já deve estar exatamente assim, já que é a configuração padrão do **`slackware`**.

Se você colocou 16 bits de cor, vá na seção "`Screen`" e duplique uma das Subsection "`Display`" (tem uma lá para 8, 16 e 32 bpp), troque o número da seção duplicada por 15 e coloque antes de todas elas, uma linha:

```
DefaultDepth 15
```

Pronto! Agora você é feliz proprietário de um X configurado e funcional!!! Tem o detalhe sutil que você vai ter que compilar um zilhão de coisas para ter uma interface gráfica interessante, mas o X em si está OK! (Se vc quiser usar o seu Mac como terminal gráfico, já pode considerar o trabalho completo).

continua...

Mantendo as Configurações

Depois de todo esse esforço, você não vai querer que todas as configurações desapareçam do dia para a noite! Ou pior, no próximo reboot!!! Para isso, o melhor a fazer é criar um `rc.modules` e colocar lá os módulos a serem carregados.

Crie um `/etc/rc.d/rc.modules` com o seguinte conteúdo:

```
----------------------------------------
# rc.modules

# Load rc.netdevice, if needed
#
if [ -x /etc/rc.d/rc.netdevice ]; then
        . /etc/rc.d/rc.netdevice
fi

# Loads the default PowerMac sound module
#
/sbin/modprobe dmasound_pmac

# And  loads serial support to PowerMacs,
# Serial ports are  the strange 9 pins
connectors
# labelled as Printer or Modem
#
/sbin/modprobe macserial
----------------------------------------
```

Torne esse arquivo executável com:

```
# chmod +x /etc/rc.d/rc.modules
```

Edite também o `/etc/rc.d/rc.S`, para que ele chame o `sysctl` no boot, configurando para você o teclado e a emulação do mouse. Faça isso colocando logo abaixo da linha "`# . /etc/rc.d/rc.serial`" as seguintes instruções:

```
----------------------------------------
# Load System Control configurations
if [ -x /sbin/sysctl ]; then
      /sbin/sysctl -p
fi
----------------------------------------
```

Claro que, para isso funcionar, será necessário um arquivo que contenha as configurações que pretendemos alterar, faça um `/etc/sysctl.conf` contendo essas linhas:

```
----------------------------------------
dev/mac_hid/keyboard_sends_linux_keycodes=1
dev/mac_hid/mouse_button_emulation=1
dev/mac_hid/mouse_button3_keycode=125
dev/mac_hid/mouse_button2_keycode=97
----------------------------------------
```

Com isso, assim que o seu computador for ligado, irá ler os módulos de som, rede e serial. O teclado e a emulação dos botões do mouse também estarão configurados.Se quiser completar o serviço, troque no `inittab` para iniciar no runlevel 4 ao invés do 3, para entrar diretamente no X.

Finalmente, o nosos Macintosh se comporta como uma máquina Linux como qualquer outra, o que é um oásis de calma e tranqüilidade. Se quiser um sistema mais completo, é hora de pegar os `SlackBuilds` do **`slackware`** e mandar ver na compilação de pacotes.

Piter PUNK <piterpk@terra.com.br>

Tabela 1 – Modos de Vídeo do Macintosh

modo	resolução	freq(Hz)	dot-clock
mac5	640x480	60	25.18MHz
mac6	640x480	67	30.00MHz
mac9	800x600	56	36.00MHz
mac10	800x600	60	40.00MHz
mac11	800x600	72	50.00MHz
mac12	800x600	75	49.50MHz
mac13	832x624	75	57.60MHz
mac14	1024x768	60	65.00MHz
mac15	1024x768	72	75.00MHz
mac16	1024x768	75	78.75MHz
mac17	1024x768	75	78.75MHz
mac18	1152x870	75	100.00MHz
mac19	1280x960	75	126.00MHz
mac20	1280x1024	75	195.00MHz
mac21	1152x768	60	-
mac22	1600x1024	60	112.27MHz

Tabela 2 – Tipos de FrameBuffer

FrameBuffer	Macintoshs
platinumfb	7200,8200
atyfb	5500, 6500, 9500, 9600, G3
valkyriefb	5200, 5400, 6200, 6400, 6360
ariel2fb	6100, 7100, 8100
controlfb	7300, 7500, 7600, 8500, 8600
chipsfb	PB2400, PB3400

continua...

Guia de Instalação do gaim-vv

O gaim-vv é um fork do Gaim que tem como objetivo dar suporte a audio e vídeo e assim permitir videoconferências como é possível no messenger da micro$oft. O site oficial do projeto é o http://gaim-vv.sourceforge.net/.

> **Obs.:**
> Esse artigo foi baseado no tutorial oficial de instalação que pode ser encontrado no site oficial do projeto. Não abordarei a configuração de webcams nesse tutorial e sim a configuração do gaim-vv para dar suporte a elas.

Veja uma screenshot dele funcionando no meu computador:

```
http://www.vivaolinux.com.br/screenshots/\
      comunidade/1098668312.snapshot13.jpg
http://www.techroot.org/modules/xcgal/\
      albums/userpics/10106/gaim-vv.jpg
```

Baixando o gaim-vv

Primeiramente vamos baixar os pacotes necessários para a instalação. Faremos a instalação de tarballs para que o tutorial sirva para qualquer distribuição.

Todos os pacotes encontram-se na pagina oficial do projeto. Os links diretos estão abaixo.

[Gaim-vv]
```
http://prdownloads.sourceforge.net/\
gaim-vv/\
gaim-0.79-vv-3.tar.gz?download
```
[Libj2k]
```
http://prdownloads.sourceforge.net/\
gaim-vv/libj2k-0.0.9.tar.gz?download
```
[Linphone-im]
```
http://prdownloads.sourceforge.net/\
gaim-vv/linphone-im.tar.gz?download
```
[LibOsIp]
```
http://simon.morlat.free.fr/\
download/0.12.2/source/
```

Pra quem quiser utilizar os pacotes prontos para **slackware** seguem os links: (Lembrando que a instalação dos pacotes é feita utilizando-se o comando "installpkg nomedopacote.tgz" como superusuário.)

```
http://l1nux.free.fr/files/10/\
      gaim-0.79-vv-i686-1.tgz
http://l1nux.free.fr/files/10/\
      libj2k-0.0.8-i686-1.tgz
http://l1nux.free.fr/files/10/\
      linphone-im-i686-1.tgz
```

Instalação dos pacotes

Lembrando que o comando para descompactar os pacotes é o "tar -zxvf pacote". Para instalar o pacote gaim-vv precisamos ter já instalados os pacotes libj2k, linphone-im e libosip. Por isso vamos fazer a instalação na seguinte ordem:

[libosip]

Faça o download and salve no seu diretório HOME, extraia (tar -zxvf...) e entre no diretório que foi criado com a descompactação. Em seguide, execute os seguintes comandos:

```
# ./configure && make";
# su
# make install
```

[libj2k]

Siga exatamente os mesmos passos da instalação do libosip.

[linphone-im]

Extraia os arquivos e entre o diretório gerado, como de costume. Em seguida, faça o seguinte:

```
# cd libr263
# make library
# cd .. && make
```

continua...

Usando o current, ocorreu um erro referente ao "docs" tanto no oRTP quanto o osipua. Para consertar esse erro foi necessário editar o arquivo "linphone-im/oRTP/Makefile" e "linphone-im/osipua/Makefile" e remover 'docs' da linha referente a SUBDIRS (por volta da linha 113 e 123 respectivamente) e depois disso executar novamente o "make".

Agora, continuando de onde paramos:

```
# su
# make install
# cp config.h \
      /usr/local/include/linphone/\
            linphone_config.h
# mkdir /usr/local/include/mediastreamer
# cp mediastreamer/*h \
      /usr/local/include/mediastreamer
```

[gaim-vv]

Faça o download and salve no seu diretorio home; extraia (tar -zxvf....) os arquivos e entre no diretório que foi criado com a descompactação. Em seguida, execute os seguintes comandos:

```
# ./configure --with-libj2k=/usr \
      --enable-linphone \
      --enable-msn-vv \
      --prefix=/usr/local && make
# su
# make install
```

Pronto! =) Estamos agora com suporte a videoconferências e conversas por áudio através do gaim-vv. Basta entrar no programa com o comando "gaim", ir em preferências e ativar o seguintes plugins: "j2k Codec" e "Linphone-vv". Agora é só clicar com o botão direito sobre o contato desejado e conferir as novas opções =)

Problemas com SSL? Veja o seguinte link: http://gaim.sourceforge.net/faq-ssl.php

É importante ressaltar que 'ainda' não é possivel enviar video para o Yahoo! Mas é perfeitamente possivel visualizar.

O Gaim-vv é um projeto em desenvolvimento e podem ocorrer erros inesperados na compilação. Nesse caso eu recomendo a participação no forum de discussão do projeto (http://sourceforge.net/forum/forum.php?forum_id=353893). Lá você encontrará respostas para a maioria das suas dúvidas. Espero que tenham gostado do artigo. Até a próxima!

r00tsh3ll <r00tsh3lll@uol.com.br>

continua...

Autores

Clayton Eduardo dos Santos, trabalha com Linux a cerca de um ano e meio e com Slackware a cerca de um ano. É matemático, mestre em Engenharia Elétrica pela USP de São Carlos e atualmente desenvolve seu projeto de pesquisa de Doutorado no Departamento de Engenharia Elétrica na USP de São Carlos, utilizando ferramentas 100% baseadas em software livre.

Deives "thefallen" Michellis, tecnólogo em Processamento de Dados pela FATEC/SP e Gerente de Desenvolvimento de Soluções Linux do Grupo GEO. Também nerd de carteirinha e ativista linux nas horas vagas.

misfit, 23 anos, bacharel em Sistemas de Informacao pelo Mackenzie/SP. LPI-I certified, linux user #215987. Participante dos grupos Linuxchix-BR e GUS-BR, contribui para o projeto linuxbr.org nas horas vagas e seus interesses incluem seguranca de redes, programacao e dispositivos embedded.

Piter PUNK, é mantenedor e principal desenvolvedor do slackpkg. Possui experiência com UNIX e Linux desde '96 tendo escrito diversos artigos em revistas da área, atualmente, trabalha como Game Designer na 3WT Corporation.

Reinaldo Nolasco Sanches (r_linux), atualmente trabalhando na Mandic LTDA no desenvolvimento de aplicações para a Web. Graduando-se em Bacharel de Sistemas de Informações pela Universidade Presbiteriana Mackenzie. Meus interesses são C/C++, Qt, Linux, Sistemas Distribuidos, Programação de Games, Inteligência Artificial, Computação Evolutiva e Heavy Metal.

toledo, iniciou com computadores em 1993 e Linux em 1998, usando Slackware 3.4 kernel 2.0.30. Desde então, vem acompanhando a evolução desse maravilhoso sistema juntamente com toda a comunidade de software livre. Atualmente editor e mantenedor do Slackware Zine.

waldemar silva junior a.k.a. R00tsh3lL, administrador de sistemas e redes Linux e programador.. graduando em Física Computacional pela UnB atualmente estagiando no ministério do planejamento trabalhando com clusters... Operador do canal #linuxajuda na brasnet, colunista no portal techroot (www.techroot.org) e certificado pela abstract (www.abstract.com.br).

Willian Ferraz a.k.a. Geek_Slack, analista de conectividade de Redes, utiliza slackware desde 1995, administra redes linux e windows desde 1999, participou da implantação arede da Praça de Atendimento da Secretaria de Finanças do Municipio de SP. Atualmente desempregado dedica-se a estudos de sistema operacional linux Slackware e Gentoo.

Construindo Aplicações Gráficas com a Qt

1. Introdução

Voltando com a segunda parte deste artigo, publicado na **slackwarezine #4**, aprofundarei-me um pouco mais no mecanismo signal/slot, mas antes vamos entrar em um assunto que não está claro para algumas pessoas da comunidade OpenSource, que é sobre a licença do Qt.

2. Licença do Qt

A toolkit Qt tem duas licenças, GNU GPL (Qt Free Edition) e Comercial (Qt Professional Edition ou Qt Enterprise Edition). O Qt edição Free pode ser usado para desenvolver aplicações livres, ou seja, sob licença GNU GPL ou similares. Para este caso, a licença do Qt será GNU GPL. Para aplicações que impõe qualquer tipo de restrição ou condição que não esteja de acordo com a licença GNU GPL ou similares, não poderá ser usado a Qt edição Free, e neste caso deverá ser usado a Qt edição comercial (Professional ou Enterprise). As aplicações com qualquer licença aprovada pela FSF.org e OpenSource.org podem usar a versão Free da Qt. Não pode ser usado a licença GNU Lesser General Public License (LGPL), pois se for permitido que aplicações não livres usem a versão Free da Qt, não há o porque da existência das edições comerciais da Qt, e a LGPL tem o seu uso desencorajado pela Free Software Foundation.

Existe a KDE Free Qt Foundation (`http://www.kde.org/whatiskde/kdefreeqtfoundation.php`), que é uma organização fundada pela Trolltech (criadora e dona da Qt) e a KDE e.V. (organização que representa o projeto KDE legalmente e financeiramente), para garantir que a Qt edição Free passe a licença BSD-style caso a Trolltech descontinue o desenvolvimento da Qt edição Free por qualquer razão, incluindo a venda da Trolltech ou falência da mesma.

A Qt edição Free está disponivel para Unix/X11, Macintosh e Linux/embedded. A Qt edição Professional e Enterprise está disponivel para Unix/X11, Macintosh, Linux/embedded e MS/Windows.

```
http://www.trolltech.com/download/\
                    opensource.html
```

3. Signals e Slots (continuação)

Como eu estava dizendo, em uma aplicação gráfica, queremos que uma mudança em um widget seja notificada a outros widgets, ou seja, queremos que qualquer tipo de objeto consiga se comunicar com outro. Por exemplo, em um parser de XML, queremos notificar uma list view, que estamos usando para mostrar a estrutura da XML, quando uma nova tag for encontrada.

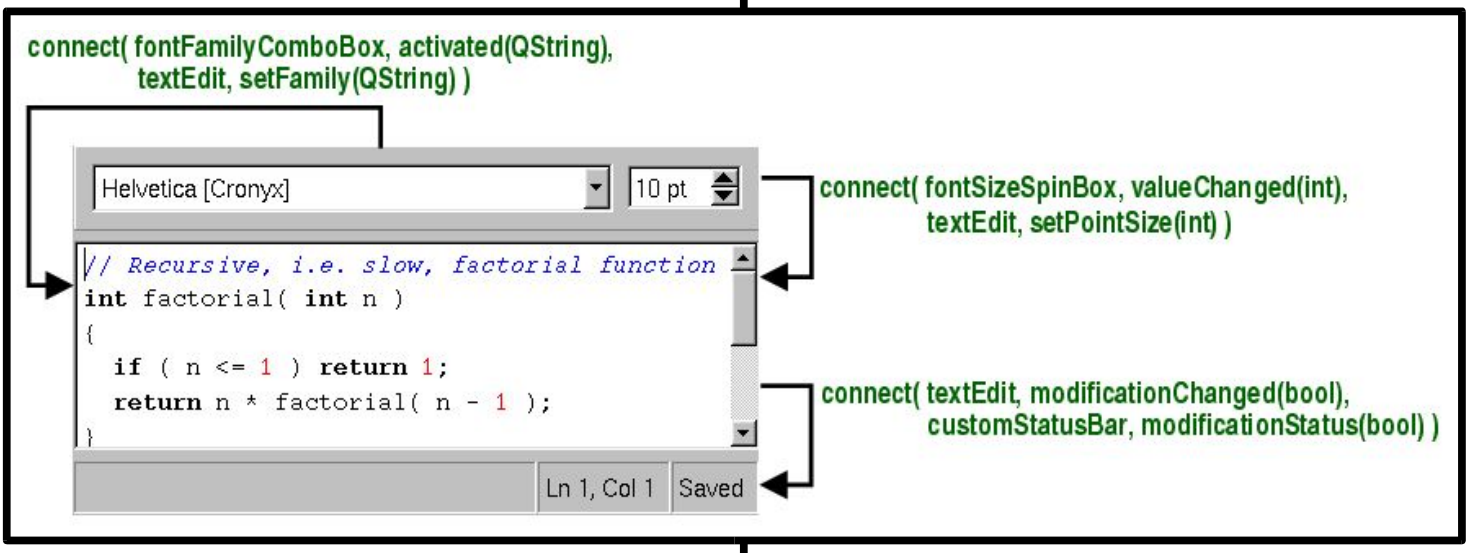
As widgets do Qt tem vários signals pré-definidos, mas pode-se criar subclasses para adicionar seus próprios signals. A Qt também tem vários slots pré-definidos, mas é uma pratica comum criar seus próprios slots para que manipulem os signals de seu interesse.

O mecanismo de signals e slots é type safe, ou seja, a assinatura de um signal deve combinar com a assinatura do slot. Mas o slot pode ter um número menor de argumentos, pois ele ignora os argumentos extras. Os signals e slots são loosely coupled (fracamente acoplados), ou seja, uma classe que emite o signal nunca vai saber qual slot receberá o sinal.

Todas as classes que herdam de QObject, ou de suas subclasses, podem conter signals e slots. Os signals são emitidos quando um objeto muda o seu estado. Isso é tudo que ele faz para se comunicar. O objeto não sabe e não se importa com quem está recebendo o sinal que ele emite. Este encapsulamento permite que este objeto seja usado como um componente de software.

continua...

continua...



Os slots podem ser usados para receber signals, mas são funções comuns. Do mesmo jeito que um objeto não sabe quem recebeu seus signals, um slot não sabe se há algum signal ligado a ele. Deste modo podem ser criados componentes independentes no Qt.

Um pequeno exemplo de uma classe em C++:

```
class Foo
{
      public:
      Foo();
      int value() const { return val; }
      void setValue( int );
      private:
      int val;
};
```

Uma pequena classe em Qt:

```
class Foo : public QObject
{
      Q_OBJECT
      public:
      Foo();
      int value() const { return val; }
      public slots:
      void setValue( int );
      signals:
      void valueChanged( int );
      private:
      int val;
};
```

Nas duas classes tem os mesmos métodos e propriedades, basicamente são a mesma classe. Mas a classe em Qt tem suporte a signals e slots que permitem notificar o mundo exterior que seu estado mudou, através do signal `valueChanged()`, e um slot `setValue()` para outros objetos enviarem signals.

Todos os objetos quem contem signals ou slots devem mencionar Q_OBJECT em sua declaração. A seguir, uma possível implementação para o `Foo::setValue( int )`:

```
void Foo::setValue( int v )
{
      if ( v != val ) {
            val = v;
            emit valueChanged(v);
      }
}
```

A linha `emit valueChanged(v)` emite o signal `valueChanged` do objeto. E como você pode ver, podemos emitir signals usando `emit signal (argumentos)`. A seguir, um exemplo de como conectar dois destes objetos:

```
Foo a, b;
connect(&a, SIGNAL(valueChanged(int)), \
            &b, SLOT(setValue(int)));
b.setValue( 11 ); // a == indefinido
                  // b == 11
a.setValue( 79 ); // a == 79
                  // b == 79
b.value();        // returns 79
```

Quando o `b.setValue(11)` foi chamado, o signal `b.valueChanged(11)` foi emitido, que não está conectado a nenhum slot. Mas quando o método `a.setValue(79)` foi chamado, `a.valueChanged (79)` também foi, e como o signal `valueChanged` de "a" está conectado ao slot `setValue` de "b", `b.setValue(79)` também é chamado.

Note que o `setValue` apenas ajusta o valor e emite o signal se `v != val`. Isso previne loops infinitos no caso de existir uma conexão cíclica, como no caso do signal `valueChanged` de "b" estivesse conectado ao slot `setValue` de "a"

continua...

continua...

Este exemplo mostra como dois objetos trabalham juntos sem conhecerem um ao outro.

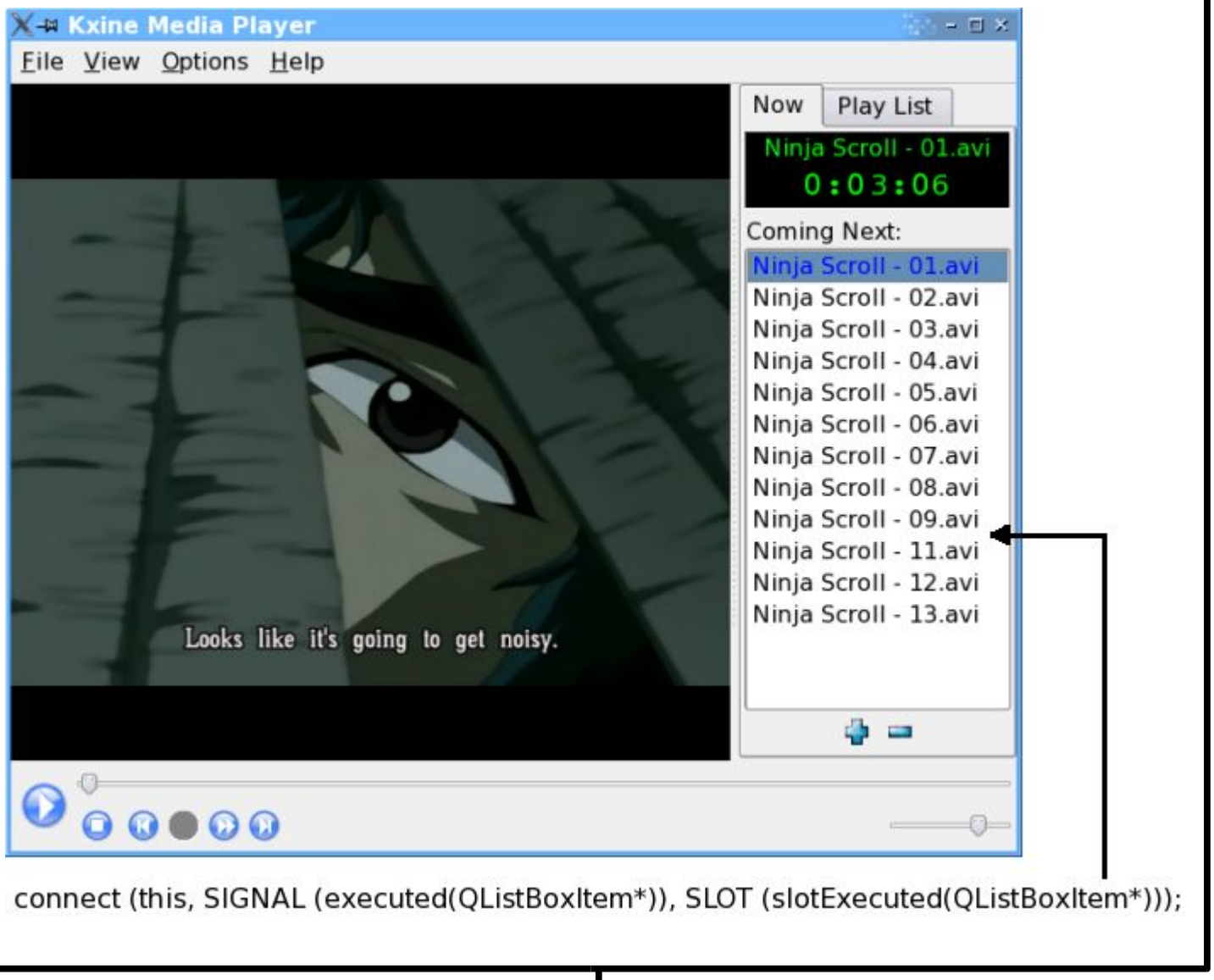
Uma pequena mudança através de pré-processadores ou a remoção das palavras signals, slots e emit faz com que o código volte a ser padrão C++.

Para criar os arquivos com código fonte para serem compilados e linkados com outros arquivos do projeto, você deve rodar o `moc` (meta object compiler) nas declarações de classes que contém signals e slots. O `moc` analiza a declaração da classe e gera um código fonte que inicializa os meta-objetos. Os meta objetos contém os nomes de todos os membros signal e slot como ponteiros para as funções.

4. Exemplo Real

Neste caso, o `KxineListBox` conecta o signal `executed()` que ele herdou da classe `KListBox` (componente do KDE) a seu slot `slotExecuted`. Quando o usuário seleciona um dos itens da lista, o signal é emitido e isso faz com que o slot `slotExecuted()` seja chamado. A seguir, a implementação do slot:

```
void KxineListBox::slotExecuted \
            (QListBoxItem *q_item)
{
      KxineListItem *item = \
            reinterpret_cast\
            <KxineListItem*>\
            (q_item);
      setPlaying (item);
      emit playMRL (item ->getMRL ());
}
```



continua...

Devemos atentar apenas a linha emit `playMRL(item ->getMRL())`, que é a linha que emite o signal `playMRL()` que mais tarde será capturado por algum outro objeto, que no nosso caso será o `KxineVideo`:

```
connect( sidebar, SIGNAL( playMRL \
      (QString)), kxine, SLOT( playFile \
      (QString)));
```

Veja que este objeto tem, através de instanciação ou ponteiro, dois objetos: o `sidebar` e o `kxine`.

Neste caso, `sidebar` é um `KxineSideBar`, que engloba todos os componentes que estão na lateral direita do `kxine`, através de instanciações de varias classes, como por exemplo a aba `Now` e a `Play List`, e nela existem duas conexões:

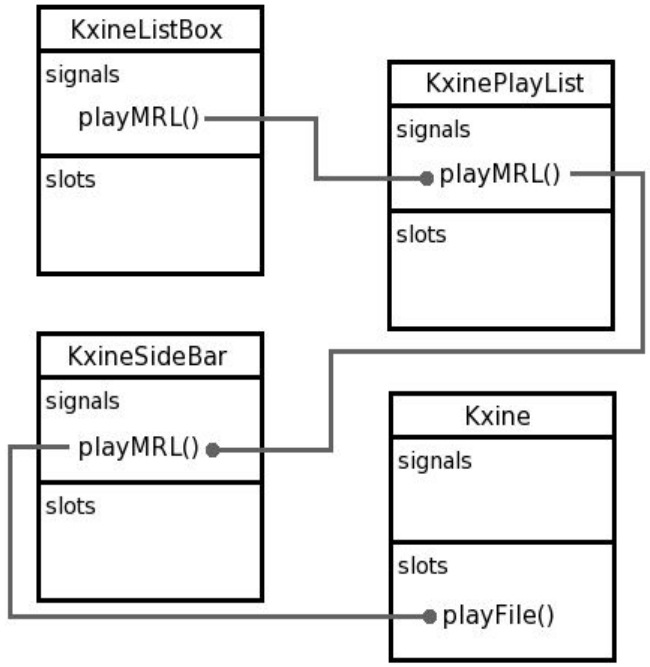
```
connect( playlist, SIGNAL( playMRL\
      ( QString ) ), SIGNAL( playMRL\
      ( QString ) ) );
connect( now, SIGNAL( playMRL\
      ( QString ) ), SIGNAL( playMRL\
      ( QString ) ) );
```

Perceba que as conexões são entre signals, os dos objetos "`playlist`", e "`now`", com os signals do `sidebar`. Em `playlist`, que é uma `KxinePlayList`, também tem uma conexão de signal para signal:

```
connect (list, SIGNAL \
      (playMRL(QString)),\
      SIGNAL (playMRL(QString)));
```

E neste caso, list é o nosso componente KxineListBox. As conexões entre signals tiveram que ser feitas, pois a aplicação vai instanciar um objeto da classe KxineSideBar sem se envolver com o resto dos objetos contidos em KxineSideBar. Quando um item no KxineListBox, que está em KxinePlayList for selecionado, este manda o signal para o KxinePlayList que manda o signals para KxineSideBar, que por fim manda o signal para KxineVideo, que ao receber este signal, chama o slot playFile() do Kxine:

```
void Kxine::playFile(QString filename)
{
      sidebar->setQueueMode();
      video->open( filename );
}
```



Tentando não me extender muito, termino aqui a explicação de mais alguns pontos sobre signals e slots e o que e como pode ser feito para que seus objetos ou componentes se comuniquem em suas aplicação.

5. Conclusão

Nesta segunda parte expliquei mais um pouco sobre signals e slots e espero que tenha suprido mais algumas dúvidas a respeito deste mecanismo de comunicação. E comecei o artigo falando um pouco sobre a licença da Qt, para mostrar que a Qt é tão livre quanto outras toolkits equivalentes.

r_linux <r_linux@yahoo.com>
misfit <k.misfit@gmail.com>

continua...

Instalando o cacti no slackware

O Cacti é um sistema de monitoramento similar ao MRTG, porém com algumas vantagens e melhor suporte ao `snmp`. Com o `cacti` você pode monitorar a quantidade de dados que trafegam na rede, monitorar o processamento, espaço em disco, entre outras coisas no servidor, além de também poder monitorar switches e roteadores.

Para uma rede com muitos hosts e equipamentos de rede é uma solução perfeita, pois traz em forma de gráficos tudo o que você precisa para um relatório bonito e com acesso web! Assim qualquer rede que você queira monitorar pode ser observada a distância.

Este documento visa orientar na instalação e configuração básicas do Cacti para ambiente Linux distribuição **`slackware`** `10.0`.

A máquina testada possuia um sistema Linux **`slackware`** `10.0`, RRDTool (requerido), SNMP (requerido), PHP(requerido), MySQL(requerido) e o próprio Cacti.

Iniciemos com a instalação das seguintes ferramentas:

RRDTool

```
http://freshmeat.net/redir/rrdtool/\
      9129/url_tgz/rrdtool-1.0.49.tar.gz
```

Após baixar este arquivo descompactamos e, dentro do diretório gerado executamos o bom e velho "`./configure ; make ; make install`"

```
# tar -xzvf rrdtool-1.0.49.tar.gz
# ./configure
# make
# make install
```

SNMP

```
http://prdownloads.sourceforge.net/\
      net-snmp/net-snmp-5.2.tar.gz
```

Executamos aqui a mesma seqüência que realizamos para o `RDDTool`, trocando apenas o nome do arquivo a ser descompactado.

PHP

```
http://br.php.net/get/php-4.3.10.tar.gz/\
      from/this/mirror
```

Após baixar o PHP iremos descompactá-lo:

```
# tar -xzvf php-4.3.10.tar.gz
```

Aqui, a seqüencia vai ser levemente alterada, pois precisamos habilitar o PHP para que tenha suporte ao SNMP. Se prepare para uma linha de comando quilométrica!

```
#./configure --with-snmp=/usr/local/ \
       --enable-ucd-snmp-hack \
       --prefix=/usr --disable-static \
       --with-apxs=/usr/sbin/apxs \
       --sysconfdir=/etc \
       --enable-discard-path \
       –-with-config-file-path=/etc/apache \
       --with-gettext=/usr/bin \
       --enable-track-vars \
       –-enable-force-cgi-redirect \
       --with-mysql --with-apxs \
       –with-snmp=/usr/local
```

Depois de executar o `./configure`, a seqüência segue a mesma:

```
# make
# make install
```

MySQL

Utilizamos o `mysql` que já veio com a distribuição, apenas levantamos o serviço da seguinte maneira:

```
# mysql_install_db
# chown -R mysql:mysql /var/lib/mysql
# chown -R mysql:mysql /var/run/mysql
# cd /etc/rc.d
# ./rc.mysql start
```

Cacti

```
http://freshmeat.net/redir/cacti/\
      20053/url_tgz/cacti-0.8.6c.tar.gz
```

Finalmente o Cacti, depois de baixarmos esse arquivo iremos descompactá-lo.

```
# tar -xzvf cacti-0.8.6c.tar.gz
```

Copiamos a pasta criada para nosso servidor de páginas que no caso é o Apache, no diretório `cacti`.

```
# cp -R cacti-0.8.6c /var/www/htdocs/cacti
```

continua...

Configurando

Depois disso, temos que criar a base de dados no mysql,

```
# mysqladmin --user=root create cacti
# mysql cacti < \
      /var/www/htdocs/cacit/cacti.sql
```

Opcional, crie um usuário e senha mysql para o cacti.

```
# mysql --user=root mysql
# GRANT ALL ON cacti.* TO \
      cactiuser@localhost \
      IDENTIFIED BY 'somepassword';
# flush privileges;
```

Depois de criado a base de dados e o usuário, temos que configurar o arquivo config.php que está localizada na pasta:

```
/var/www/htdocs/cacti/include
```

Neste nosso exemplo. As seguintes linhas devem ser alteradas para aquilo que você selecionou nas opções acima.

```
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "cacti";
```

Agora damos as permissões apropriadas para os diretórios rra/ e log/

```
# chown -R cactiuser rra/ log/
```

Terminada a configuração do arquivo, colocamos a seguinte linha no crontab.

```
*/5 * * * * cactiuser php \
      /var/www/htdocs/cacti/cmd.php \
      > /dev/null 2>&1
```

(Entre com um nome de usuário válido no lugar de cactiuser este usuário será utilizado no próximo passo da configuração).

SNMP – Para configurar o snmp de acordo com o que exatamente vc necessita, vc pode executar o arquivo snmpconf,onde lhe será feita uma série de perguntas que vc pode responder, não vou me estender aqui, devido as inúmeras funcionalidades do SNMP, colocarei um exemplo da utilização mais básica do snmp.

No diretório onde está instalado o SNMP crie um arquivo chamado snmpd.conf com as seguintes linhas.

```
syslocation     "SNMP do Geek_Slack"
syscontact      geek_slack@unitednerds.org
rwcommunity     private
rocommunity     public
authtrapenable  1
trapcommunity   public
trapsink        localhost
trap2sink       localhost
```

Analisando cada linha teremos:

syslocation: Um nome dado como referencia para o computador em questão.

syscontact: Um endereço de e-mail para contato.

Essas duas linhas são opcionais, mas é bastante interessante mantê-las com informações úteis principalmente porque em redes com muitos computadores, o monitoramento fica mais ágil e a resolução do problema também.

rwcommunity: comunidade que poderá ler e escrever, ou seja, quem pode executar os processos e gerenciar a máquina.

rocommunity: comunidade que poderá ver as tabelas e os gráficos sem poder alterar absolutamente nada no host.

authtrapenable: nível de authoridade, na verdade quer dizer qual versão do SNMP está sendo utilizada.

trapcommunity: comunidade da trap a ser lida.

trapsink: Host para onde será enviado a trap.

trap2sink: Host para onde será enviado a trap (SNMPv2).

Agora crie um diretório onde o cacti irá criar as tabelas e gráficos para monitoramento:

```
# mkdir /var/www/htdocs/monitor
# chown -R cactiuser:root \
            /var/www/htdocs/cacti
# chown -R cactiuser:root \
            /var/www/htdocs/monitor
```

Finalmente é hora de REALMENTE instalar o cacti, aponte o navegador de sua preferencia para: http://localhost/cacti

Siga as instruções da tela e certifique-se de colocar os paths CORRETOS para cada programa solicitado no cacti. Após tudo isso basta vc adicionar as máquinas a serem monitoradas e pronto... você tem um completo relatório de cada máquina da rede!

Geek_Slack <willian@spbn.com.br>

continua...

Compilando um kernel para PowerMac

Como tive alguma dificuldade para saber o era necessário ter no kernel para PPC no caso do meu PowerMac, aqui vai uma pequena lista do que compilar e, para que serve cada coisa. Claro, você pode selecionar alguns itens a mais ou a menos, mas em essência é isso aqui -;). Se você possui um NewWorld Mac, lembre de habilitar o USB, sob o risco de ficar com uma máquina inusável. Boa Sorte!

```
# cd /usr/src/linux
# make menuconfig
```

Agora vamos navegar pelo menu e selecionar o que acharmos interessante:

```
SCSI support
SCSI low-level drivers
      <*> MESH (PM internal SCSI) support
      <*> 53C94 (PM external SCSI) support
```

O CDROM do seu Mac deve ser SCSI, bem como os seus periféricos "adicionais", como scanners, zip drive, etc... Logo, é importante habilitar o suporte a esses dispositivos, tanto os internos (MESH) como os externos (53C94).

```
Network device support
      [*] Ethernet (10 or 100Mbit)
      <M> MACE (PM ethernet) support
      <M> BMAC (G3 ethernet) support
      [*] EISA, VLB, PCI and on board \
                        controllers
      <M> Realtek RTL-8139 PCI Fast \
                  Ethernet Adapter support
```

Se o seu Mac possui rede on-board, grandes chances de ser um MACE. Selecionei o BMAC também porque não sei se o Beige G3 usa ele. De qualquer modo, ambos estão como módulos. Se você tem uma placa de rede PCI (como eu) deveselecioná-la também.

```
Console drivers
Frame-buffer support
      [*] Open Firmware frame buffer \
                        device support
      [*] Apple "control" display support
      [*] Apple "platinum" display support
      [*] Apple "valkyrie" display support
      [*] Chips 65550 display support
      [*] ATI Mach64 display support
      [*] ATI Rage128 display support
```

Você precisa de suporte a pelo menos um framebuffer. Mas não de todos. Como você está compilando o kernel especificamente para a sua máquina, pode selecionar apenas o framebuffer que for usar.

```
Input core support
```

Deixe tudo, menos o suporte a joystick selecionado.

```
Macintosh device drivers
      [*] Support for CUDA based PM
      [*] Support for PMU based PM
      [*] Support for PM floppy
      <M> Support for PM serial ports
      [*] Apple Desktop Bus (ADB) support
      [*] Include MacIO (CHRP) ADB driver
      [*] Use input layer for ADB devices
      [*] Support for ADB raw keycodes
      [*] Support for mouse button 2+3 \
                              emulation
```

Praticamente todos os hardwares engraçados e o suporte a diversos xunchos estão todos dentro de uma mesma área -:). Selecione praticamente tudo. Sem ADB você fica sem teclado e mouse. Muito cuidado nessa hora.

```
Sound
      [M] Sound card support
      [M] PM DNA sound support
```

Acho que todos os Macintosh possuem som on-board. E esse é o suporte a som que você precisa -:). Acho que se você colocar uma placa de som PCI, os módulos "normais" devem funcionar também.

Agora, depois de tudo selecionado, execute a seguinte seqüência (e tenha bastante paciência...):

```
# make dep
# make clean
# make vmlinux
# make modules
# make modules_install
```

No final, será gerado um arquivo `vmlinux` no `/usr/src/linux`. Copie esse arquivo para o seu `/boot` e execute novamente o `quik`. E finito! -:)

Piter PUNK <piterpk@terra.com.br>