Adicionando capacidade Multilib ao Slackware na arquitetura x86_64

Este artigo contém instruções sobre como criar um Slackware64 *verdadeiramente multilib*. Um sistema Linux 64 bits multilib é capaz de rodar programas de 64 bits e 32 bits. O Filesystem Hierarchy Standard descreve o método ideal para obter uma separação clara entre os softwares de 64 bits e 32 bits em um único sistema. Ao iniciar o desenvolvimento do "Slackware64" (o porte oficial para a arquitetura Wx86_64) optamos por adotar este padrão. Portanto, o Slackware64 foi configurado para procurar bibliotecas de 64 bits nos diretórios `/lib64` e `/usr/lib64`. É por isso que chamo o Slackware64 de "multilib-ready" (multilib-preparado)- embora as bibliotecas de 32 bits sejam procuradas em `/lib` e `/usr/lib`, o Slackware64 não vem com nenhum software de 32 bits. Há mais um passo que deve ser executado (por você, o usuário) antes que o Slackware64 possa ser chamado de "multilib-enabled" (multilib-habilitado).

Isso é feito da seguinte maneira:

1. Primeiro, precisamos mudar para as versões multilib de
   - *glibc* (ou seja, uma glibc que suporta a *execução* de binários de 32 e 64 bits) e
   - *gcc* (ou seja, capaz de *compilar* binários de 32 bits, bem como binários de 64 bits).
2. Em seguida, as bibliotecas do sistema são retiradas do Slackware 32 bits e instaladas no sistema Slackware de 64 bits junto com suas versões de 64 bits, o que completa o processo de criação de uma camada de software de compatibilidade de 32 bits.

O Slackware64 tem uma vantagem sobre outros "forks" de 64 bits que existem. Esses forks criam a camada de compatibilidade com sistema de 32 bits recompilando grande parte de seus pacotes para 32 bits. Por outro lado, o Slackware é uma distribuição que possui uma versão para 32 bits e 64 bits, em que cada versão é desenvolvida de forma paralela. Isto quer dizer que você não precisa compilar todos os pacotes de 32 bits a partir do zero para que seja possível adicionar a camada de compatibilidade ao sistema 64 bits. Você simplesmente pega os pacotes da árvore de pacotes do Slackware 32 bits! \\Essa é uma das razões para não adicionar todos os pacotes multilib ao Slackware64 - nós criamos as condições adequadas, mas é necessário que o usuário atue caso necessite da compatibilidade.
Em uma seção posterior, eu vou explicar como você pode buscar um pacote do Slackware 32 bits (digamos, o pacote "mesa") e reempacotá-lo em um pacote "mesa-compat32" que pode ser instalado diretamente no Slackware64.

Slackware para a arquitetura x86_64 (ou "*Slackware64*") é um sistema operacional puro de 64 bits, porém pode ser facilmente atualizado para a multlib. *Originalmente, o Slackware64 só é capaz de compilar e executar programas de 64 bits.*

Vantagens de um sistema multilib

Vou dar alguns exemplos de programas que necessitam do multilib em um Slackware de 64 bits, porque eles não serão inciados ou compilado no Slackware64 sem uma camada de compatibilidade 32 bits:

- Wine
  A maioria dos programas Windows ainda são 32 bits, e para executar esses programas no Linux com o Wine, você precisa de uma versão de 32 bits do Wine.
- VirtualBox
  O popular software de máquinas virtuais. Embora seja (parcialmente) de código-aberto, ele ainda precisa de bibliotecas de compatibilidade de 32 bits no Slackware de 64 bits.
- Steam
  A plataforma de jogos super popular ainda precisa de um client 32 bits. A maioria dos jogos disponíveis também são de 32 bits.
- Skype, client do Citrix,...
  Esses programas são proprietários e de código fechado. Dependemos do desenvolvedor para disponibilizar os binários de 64 bits. Até agora, isso não aconteceu com esses programas exemplificados.

Felizmente, o suporte a 64 bits está se tornando cada vez mais comum. A Adobe foi um ponto sensível por muito tempo, mas eles finalmente lançaram seu plugin do navegador Flash em uma versão de 64 bits. A Sun (agora absorvida pela Oracle) revelou uma versão de 64 bits de seu plugin de navegador Java. Esses dois eventos foram os principais gatilhos para começar a trabalhar no Slackware64.

Obtendo os pacotes multilib

Você pode baixar um conjunto de pacotes e scripts para multilib-enabled no meu site: “http://slackware.com/~alien/multilib”.

Além de vários arquivos README (este artigo Wiki basicamente é uma versão aprimorada de um desses READMEs), você encontrará um subdiretório para cada versão do Slackware de 64 bits abaixo do diretório superior “*multilib*”. Existe outro diretório chamado “source”. O diretório “*source*” contém as fontes de pacotes e scripts SlackBuild.

O que você realmente está interessado - os pacotes binários - está disponível no diretório com o número da versão do Slackware, abaixo do diretório de nível superior. Cada diretório também contém um subdiretório “*slackware64-compat32*” onde você encontrará um conjunto essencial de pacotes Slackware de 32 bits convertidos, prontos para instalação em seu Slackware de 64 bits.

Mantendo seu multilib atualizado

Para se manter atualizado, aconselho você a ficar de olho no ChangeLog (RSS feed) que mantenho para meus pacotes multilib. Normalmente, terei os pacotes *glibc e gcc atualizados* disponíveis um dia após o Slackware ter atualizações para gcc e glibc.

Automatização:

1. Confira o compat32pkg, por Sébastien Ballet, que automatiza este processo de forma semelhante ao slackpkg.
2. Se você preferir o slackpkg para gerenciamento de pacotes, vale a pena conferir o slackpkg+, uma extensão do slackpkg que gerencia os pacotes que você instalou de repositórios de terceiros - incluindo multilib. Quando configurado corretamente, você poderá manter seu

multilib facilmente executando:

```
# slackpkg update
# slackpkg upgrade multilib
# slackpkg install multilib
```

Esse último comando mostrará se algum pacote novo foi adicionado à coleção de pacotes "compat32", como llvm-compat32 e orc-compat32 recentemente.

- Esta é a aparência de uma configuração típica - para um computador executando o Slackware-current e usando o repositório de testes KDE do AlienBOB. O PKGS_PRIORITY garante que os pacotes multilib do gcc e glibc tenham precedência sobre os originais do Slackware. A palavra-chave "multilib", que define o nome do repositório, deve ser a mesma palavra-chave usada nos comandos "slackpkg" acima. A escolha da palavra "multilib" é arbitrária, poderia muito bem ser "compat32", desde que seja usada de forma consistente.
  O conteúdo de um arquivo de exemplo "/etc/slackpkg/slackpkgplus.conf" seria o seguinte:

```
SLACKPKGPLUS=on
VERBOSE=1
ALLOW32BIT=off
USEBL=1
WGETOPTS="--timeout=5 --tries=1"
GREYLIST=on
PKGS_PRIORITY=( multilib restricted alienbob ktown )
REPOPLUS=( slackpkgplus multilib restricted alienbob ktown )
MIRRORPLUS['multilib']=http://bear.alienbase.nl/mirrors/people/alien/multilib/current/
MIRRORPLUS['alienbob']=http://bear.alienbase.nl/mirrors/people/alien/sbrepos/current/x86_64/
MIRRORPLUS['restricted']=http://bear.alienbase.nl/mirrors/people/alien/restricted_sbrepos/current/x86_64/
MIRRORPLUS['ktown']=http://bear.alienbase.nl/mirrors/alien-kde/current/latest/x86_64/
MIRRORPLUS['slackpkgplus']=http://slakfinder.org/slackpkg+/
```

Ativando o Suporte multilib no Slackware64

Guia rápido

Esta seção contém as instruções essenciais para adicionar capacidade multilib completa ao seu sistema Slackware64. Se você deseja entender o processo em mais detalhes, ou precisa de informações sobre como compilar software de 32 bits no Slackware64, você também deve ler as seções a seguir.
Observe que o "#" na frente dos comandos representa um *prompt de root*.

- Baixe os pacotes do meu site (eu dei a você a URL na seção anterior, mas este exemplo está usando uma URL de um mirror). Suponha que você esteja executando o Slackware 14.2. Você

executa:

```
# SLACKVER=14.2
# mkdir multilib
# cd multilib
# lftp -c "open http://bear.alienbase.nl/mirrors/people/alien/multilib/
; mirror -c -e ${SLACKVER}"
# cd ${SLACKVER}
```

- Atualize seus pacotes "*gcc*" e "*glibc*" do Slackware de 64 bits para minhas versões multilib. Execute o comando

```
# upgradepkg --reinstall --install-new *.t?z
```

depois de mudar para o diretório onde você baixou esses pacotes.
Este comando também instalará um pacote adicional chamado "*compat32-tools*".

- Se você também baixou um diretório chamado *slackware64-compat32* (meu exemplo de comando "`lftp`" terá feito isso), então você está com sorte, porque eu já fiz a conversão do pacote de 32 bits para você! Tudo o que você faz é executar o comando:

```
# upgradepkg --install-new slackware64-compat32/*-compat32/*.t?z
```

  que irá instalar todos os pacotes Slackware de 32 bits convertidos (ou atualizá-los se você já instalou pacotes multilib mais antigos, por exemplo, quando você estiver atualizando para um Slackware mais novo). Isso é tudo!

- Se você não conseguir encontrar um subdiretório chamado *slackware64-compat32* , então você não o baixou ou o espelho de download não o forneceu. Nesse caso, você mesmo deve fazer a conversão do pacote de 32 bits. Não é nada difícil, leva alguns minutos:
  - O mais rápido é se você tiver um diretório local com pacotes originais do Slackware de 32 bits disponíveis (também chamado de *local mirror*). Quem comprou um DVD oficial do Slackware pode simplesmente usar esse DVD: ele tem dois lados e o Slackware de 32 bits está em um dos lados. Para o propósito deste exemplo assumirei que você tem uma árvore de diretórios local do Slackware de 32 bits disponível em "`/home/ftp/pub/slackware/slackware-14.2/slackware/`".
    Deve haver subdiretórios chamado 'a', 'ap', 'd', 'l', 'n', 'x' imediatamente abaixo deste diretório. (Se você montou um DVD do Slackware, seu diretório provavelmente será "`/media/SlackDVD/slackware/`", mas não vou usar isso nos comandos de exemplo abaixo).
  - Crie um novo diretório vazio (vamos chamá-lo de 'slackware64-compat32') e mude para ele:

```
# mkdir slackware64-compat32 ; cd slackware64-compat32
```

  - Execute o seguinte comando para criar um conjunto de pacotes de compatibilidade de 32 bits, usando o diretório para os pacotes oficiais do Slackware de 32 bits como entrada:

```
# massconvert32.sh -i
/home/ftp/pub/slackware/slackware-14.2/slackware/
```

  - A etapa anterior demora um pouco. Quando terminar, instale os 90 MB de pacotes

Slackware de 32 bits recém-convertidos que foram criados em subdiretórios abaixo do *diretório atual*:

```
# upgradepkg --install-new *-compat32/*.t?z
```

- Feito! Agora você pode começar a baixar, instalar e executar programas de 32 bits. Não foi tão difícil, foi?

Se você usar um gerenciador de pacotes como *slackpkg* nas versões mais antigas do Slackware que 13.37, você terá que adicionar todos os nomes de pacotes `glibc` e `gcc` à blacklist de pacotes. Se você não tomar este cuidado, você corre o risco de seu gerenciador de pacotes substituir acidentalmente suas versões multilib pelas versões originais de 64 bits do Slackware!
Se você rodar o Slackware 13.37 ou mais recente, então o slackpkg suporta expressões regulares no arquivo de blacklist. Nesse caso, uma única linha em `/etc/slackpkg/blacklist` será suficiente para colocar todos os meus pacotes na blacklist (incluindo pacotes multilib gcc e glibc e todos os pacotes `compat32`):

```
[0-9]+alien
[0-9]+compat32
```

Por outro lado, se você está usando a extensão slackpkg chamada slackpkg+ então você definitivamente **não** deve colocar esses pacotes na lista negra, porque isso impede o slackpkg+ de gerenciá-los!

Se você estiver executando o Slackware 13.1 ou mais recente e baixou o pacote compat32-tools para essa versão, o script *massconvert32.sh* pode usar um servidor remoto para baixar os pacotes do Slackware de 32 bits, em vez de exigir um Slackware local mirror ou um DVD. Use o parâmetro "-u" para especificar o URL remoto dessa forma:

```
# massconvert32.sh -u
http://algumservidor.org/path/para/slackware-14.2/slackware
```

Instruções detalhadas

Atualizando glibc e gcc

Os seguintes pacotes glibc/gcc são substituições para - *não adições aos* - pacotes padrão do Slackware. Use o programa "`upgradepkg`" para atualizar para minhas versões multilib do gcc e glibc. Você precisará deles para executar (glibc) e construir (gcc) software de 32 bits em seu computador Slackware de 64 bits:

Slackware64 13.0

- O pacote de compiladores gcc:
  - gcc-4.3.3_multilib-x86_64-4alien.txz
  - gcc-g++-4.3.3_multilib-x86_64-4alien.txz
  - gcc-gfortran-4.3.3_multilib-x86_64-4alien.txz
  - gcc-gnat-4.3.3_multilib-x86_64-4alien.txz
  - gcc-java-4.3.3_multilib-x86_64-4alien.txz
  - gcc-objc-4.3.3_multilib-x86_64-4alien.txz
- As bibliotecas GNU libc:
  - glibc-2.9_multilib-x86_64-3alien.txz
  - glibc-i18n-2.9_multilib-x86_64-3alien.txz
  - glibc-profile-2.9_multilib-x86_64-3alien.txz
  - glibc-solibs-2.9_multilib-x86_64-3alien.txz
  - glibc-zoneinfo-2.9_multilib-noarch-3alien.txz

Slackware64 13.1

- O pacote de compiladores gcc:
  - gcc-4.4.4_multilib-x86_64-1alien.txz
  - gcc-g++-4.4.4_multilib-x86_64-1alien.txz
  - gcc-gfortran-4.4.4_multilib-x86_64-1alien.txz
  - gcc-gnat-4.4.4_multilib-x86_64-1alien.txz
  - gcc-java-4.4.4_multilib-x86_64-1alien.txz
  - gcc-objc-4.4.4_multilib-x86_64-1alien.txz
- As bibliotecas GNU libc:
  - glibc-2.11.1_multilib-x86_64-3alien.txz
  - glibc-i18n-2.11.1_multilib-x86_64-3alien.txz
  - glibc-profile-2.11.1_multilib-x86_64-3alien.txz
  - glibc-solibs-2.11.1_multilib-x86_64-3alien.txz
  - glibc-zoneinfo-2.11.1_multilib-noarch-3alien.txz

Slackware64 13.37

- O pacote de compiladores gcc:
  - gcc-4.5.2_multilib-x86_64-2alien.txz
  - gcc-g++-4.5.2_multilib-x86_64-2alien.txz
  - gcc-gfortran-4.5.2_multilib-x86_64-2alien.txz
  - gcc-gnat-4.5.2_multilib-x86_64-2alien.txz
  - gcc-java-4.5.2_multilib-x86_64-2alien.txz
  - gcc-objc-4.5.2_multilib-x86_64-2alien.txz
- As bibliotecas GNU libc:
  - glibc-2.13_multilib-x86_64-7alien.txz
  - glibc-i18n-2.13_multilib-x86_64-7alien.txz
  - glibc-profile-2.13_multilib-x86_64-7alien.txz
  - glibc-solibs-2.13_multilib-x86_64-7alien.txz

Slackware64 14.0

- O pacote de compiladores gcc:
  - gcc-g++-4.7.1_multilib-x86_64-1alien.txz
  - gcc-gfortran-4.7.1_multilib-x86_64-1alien.txz
  - gcc-gnat-4.7.1_multilib-x86_64-1alien.txz
  - gcc-go-4.7.1_multilib-x86_64-1alien.txz
  - gcc-java-4.7.1_multilib-x86_64-1alien.txz
  - gcc-objc-4.7.1_multilib-x86_64-1alien.txz
- As bibliotecas GNU libc:
  - glibc-2.15_multilib-x86_64-9alien.txz
  - glibc-i18n-2.15_multilib-x86_64-9alien.txz
  - glibc-profile-2.15_multilib-x86_64-9alien.txz
  - glibc-solibs-2.15_multilib-x86_64-9alien.txz

Slackware64 14.1

- O pacote de compiladores gcc:
  - gcc-4.8.2_multilib-x86_64-1alien.txz
  - gcc-g++-4.8.2_multilib-x86_64-1alien.txz
  - gcc-gfortran-4.8.2_multilib-x86_64-1alien.txz
  - gcc-gnat-4.8.2_multilib-x86_64-1alien.txz
  - gcc-go-4.8.2_multilib-x86_64-1alien.txz
  - gcc-java-4.8.2_multilib-x86_64-1alien.txz
  - gcc-objc-4.8.2_multilib-x86_64-1alien.txz
- As bibliotecas GNU libc:
  - glibc-2.17_multilib-x86_64-10alien.txz
  - glibc-i18n-2.17_multilib-x86_64-10alien.txz
  - glibc-profile-2.17_multilib-x86_64-10alien.txz
  - glibc-solibs-2.17_multilib-x86_64-10alien.txz

Slackware64 14.2

- O pacote de compiladores gcc:
  - gcc-5.3.0_multilib-x86_64-3alien.txz
  - gcc-g++-5.3.0_multilib-x86_64-3alien.txz
  - gcc-gfortran-5.3.0_multilib-x86_64-3alien.txz
  - gcc-gnat-5.3.0_multilib-x86_64-3alien.txz
  - gcc-go-5.3.0_multilib-x86_64-3alien.txz
  - gcc-java-5.3.0_multilib-x86_64-3alien.txz
  - gcc-objc-5.3.0_multilib-x86_64-3alien.txz
- As bibliotecas GNU libc:
  - glibc-2.23_multilib-x86_64-2alien.txz
  - glibc-i18n-2.23_multilib-x86_64-2alien.txz
  - glibc-profile-2.23_multilib-x86_64-2alien.txz
  - glibc-solibs-2.23_multilib-x86_64-2alien.txz

Slackware64 current

- Contanto que você não veja um diretório separado chamado “*current*”, você pode apenas usar os arquivos no diretório para a versão estável mais recente.
- O pacote de compiladores gcc:
  - gcc-7.1.0_multilib-x86_64-2alien.txz
  - gcc-brig-7.1.0_multilib-x86_64-2alien.txz
  - gcc-g++-7.1.0_multilib-x86_64-2alien.txz
  - gcc-gfortran-7.1.0_multilib-x86_64-2alien.txz
  - gcc-gnat-7.1.0_multilib-x86_64-2alien.txz
  - gcc-go-7.1.0_multilib-x86_64-2alien.txz
  - gcc-objc-7.1.0_multilib-x86_64-2alien.txz
- As bibliotecas GNU libc:
  - glibc-2.25_multilib-x86_64-3alien.txz
  - glibc-i18n-2.25_multilib-x86_64-3alien.txz
  - glibc-profile-2.25_multilib-x86_64-3alien.txz
  - glibc-solibs-2.25_multilib-x86_64-3alien.txz

Desde a atualização para o `gcc 7`, não há mais o pacote `gcc-java` porque seu desenvolvimento foi interrompido.

O pacote `glibc-zoneinfo` não faz parte do multilib, pois não contém código. Você precisa instalar o pacote `glibc-zoneinfo` do Slackware.

Todas as versões do Slackware

Existe um pacote adicional que você precisa instalar usando o programa “installpkg”. A versão pode variar para cada lançamento do Slackware, mas o pacote pode ser encontrado no mesmo diretório onde você também encontra as versões multilib de `gcc` e `glibc`:

- O “32bit toolkit” (scripts que facilitam a criação de pacotes de 32 bits)
  - compat32-tools-3.7-noarch-1alien.tgz

Adicionando bibliotecas 32-bit Slackware

A atualização de glibc e gcc que descrevi na seção anterior muda seu sistema de “ *multilib-ready* ” para “ *multilib-enabled* ”.
Agora, tudo que você precisa fazer é instalar as versões 32 bits dos softwares do Slackware para que os futuros programas de 32 bits que você vai instalar e/ou compilar encontrem todas as bibliotecas de 32 bits de que precisam para funcionar.

Isso não é tão simples quanto pegar pacotes de 32 bits do Slackware e instalá-los no Slackware64: * Em primeiro lugar, você acabará com vários pacotes com o mesmo nome (dois pacotes 'mesa', dois pacotes 'zlib', etc …) o que será confuso para você e também para o *slackpkg* gerenciador de pacotes. * Além disso, se o pacote de 32 bits contiver binários (algo como `/usr/bin/foo`), eles

sobrescreverão suas contrapartes de 64 bits quando você instalar o pacote de 32 bits. Vai bagunçar seriamente o seu sistema se isso acontecer.

Um pouco de cuidado extra é necessário para que arquivos desnecessários/indesejados sejam retirados dos pacotes de 32 bits antes de você instalá-los. O que você precisa é de um pacote de 32 bits que não entre em conflito com o que já está presente no Slackware de 64 bits. Daí o nome "pacote de compatibilidade de 32 bits".

Decidi que seria um desperdício de banda de download se eu mesmo criasse versões de compatibilidade de 32 bits dos pacotes do Slackware. Afinal, você provavelmente comprou o DVD do Slackware 14.2, então já possui as versões de 64 e 32 bits do Slackware … ou então a árvore do Slackware de 32 bits está disponível para download gratuito, é claro 😉

Em vez disso, escrevi alguns scripts (partes do código do script foram escritas por Fred Emmott da para o Slamd64) e os envolvi em um pacote "*compat32-tools*". Seu objetivo é permitir que você extraia o conteúdo de qualquer pacote do Slackware de 32 bits e use-o para criar um novo pacote que possa ser instalado com segurança em seu Slackware de 64 bits.

Este pacote "*compat32-tools*" precisa de alguma explicação.
Por favor, leia o arquivo '*README*' detalhado no diretório `/usr/doc/compat32-tools - * /`, ele irá ajudá-lo. Estes são os três scripts úteis que o pacote instala:

- */etc/profile.d/32dev.sh*
  Este é o mesmo script que vem com o Slamd64. Ele reconfigura seu ambiente de shell para que seja mais fácil para você compilar o software de 32 bits (preferindo os compiladores e bibliotecas de 32 bits em vez de suas versões de 64 bits)

- *convertpkg-compat32*
  Este script pega um pacote Slackware de 32 bits e o converte em um pacote '-compat32' que você pode instalar com segurança (usando "installpkg") no Slackware64, junto com a versão de 64 bits do mesmo pacote de software. Por exemplo: suponha que você precise de bibliotecas de 32 bits que estão no pacote mesa. Você pega o pacote mesa do Slackware de 32 bits (`x/mesa-7.5-i486-1.txz`) e executa

  ```
  # convertpkg-compat32 -i /path/to/mesa-7.5-i486-1.txz
  ```

  que criará um novo pacote chamado `mesa-compat32-7.5-x86_64-1compat32.txz`. Este novo pacote (que é criado em seu diretório `/tmp` a menos que você especifique outro destino) é basicamente o antigo pacote de 32 bits, mas retirado de coisas não essenciais. O nome de base alterado (*mesa* torna-se *mesa-compat32*) permite que você instale este novo pacote no Slackware64 onde ele coexistirá com o pacote de 64 bits *mesa*, sem sobrescrever nenhum arquivo.
  O script deixa arquivos temporários no diretório "`/tmp/package-<prgnam>-compat32`" que você pode deletar com segurança.

- *massconvert32.sh*
  Este script contém uma lista interna do que considero o subconjunto essencial dos pacotes de 32 bits do Slackware. Ele usa o script "*convertpkg-compat32*" acima para pegar todos os pacotes que estão nesta lista interna e convertê-los em pacotes '-compat32'.
  Você precisa executar este script apenas uma vez, por exemplo como este (o exemplo pressupõe que você montou seu DVD do Slackware de 32 bits em `/mnt/dvd`):

```
# massconvert32.sh -i /mnt/dvd/slackware -d ~/compat32
```

Esta ação resultará em cerca de 150 MB de novos pacotes que você encontrará dentro do diretório recém-criado `~/compat32` (o nome do diretório é arbitrário, é claro, eu o escolhi por causa deste exemplo). Esses pacotes incluem o componente de 32 bits do seu sistema multilib Slackware64.
Eles devem ser instalados usando "`installpkg`" e fornecem uma camada de compatibilidade de 32 bits bastante completa sobre o Slackware64:

```
# installpkg~/compat32/*/*.t?z
```

Se você estiver atualizando de uma versão anterior desses pacotes (porque, por exemplo, você atualizou seu Slackware de 64 bits para uma versão mais recente) então você não usa "`installpkg`" é claro, mas "`upgradepkg –install-new` ":

```
# upgradepkg --install-new ~/compat32/*/*.t?z
```

O parâmetro "*–install-new*" é necessário para instalar os novos pacotes *compat32* que foram adicionados entre as versões.

Ao instalar os pacotes *compat32* você notará que alguns irão mostrar erros sobre arquivos ausentes em `/etc`. Isso ocorre "por definição" e esses erros podem ser ignorados. Essas mensagens são causadas pelo fato de que os arquivos em `/etc` são removidos de um pacote "-compat32" durante a conversão (exceto para *pango* e *gtk+2*). Presumo que os arquivos em `/etc` já terão sido instalados pelos pacotes originais de 64 bits.
Um exemplo desses "erros" para o pacote `cups-compat32`:

```
Executing install script for cups-compat32-1.3.11-x86_64-1.txz.
install/doinst.sh: line 5: [: too many arguments
cat: etc/cups/interfaces: Is a directory
cat: etc/cups/ppd: Is a directory
cat: etc/cups/ssl: Is a directory
cat: etc/cups/*.new: No such file or directory
cat: etc/dbus-1/system.d/cups.conf.new: No such file or
directory
chmod: cannot access `etc/rc.d/rc.cups.new': No such file or
directory
cat: etc/rc.d/rc.cups.new: No such file or directory
Package cups-compat32-1.3.11-x86_64-1.txz installed.
```

Se você está pensando em usar o script `convertpkg-compat32` para converter um pacote **não-Slackware** em um pacote *-compat32*, eu devo aconselhar fortemente a não fazê-lo. O script foi escrito com um único propósito: disponibilizar versões de 32 bits dos binários/bibliotecas oficiais do Slackware64 em uma configuração multilib. Como tal, o script removerá muitas coisas que estão presentes no pacote original de 32 bits - coisas que se espera tenham sido instaladas como parte da versão de 64 bits

do pacote.
Em quase todos os casos em que você baixou um pacote de 32 bits que não é do Slackware e quer fazê-lo funcionar no Slackware64, a melhor maneira é encontrar as fontes e construir uma versão de 64 bits do pacote. Alternativamente, apenas *instale o* pacote original de 32 bits em vez de tentar "convertê-lo" e, em seguida, execute-o na linha de comando para descobrir qualquer biblioteca de 32 bits ausente que você ainda possa ter que extrair de um pacote oficial do Slackware.

Executando programas 32 bits

Executar um programa pré-compilado de 32 bits é fácil depois de você ter feito a preparação do sistema descrita acima. Basta fazer o download, instalar e executar!

Às vezes, você pode se deparar com um programa que requer uma determinada biblioteca do Slackware de 32 bits que você ainda não tem disponível. Nesse caso, descubra qual pacote do Slackware de 32 bits contém essa biblioteca faltante. Use o script "*convertpkg-compat32*" para converter aquele pacote Slackware original de 32 bits e instale o pacote "*compatível*" de 32 bits resultante no Slackware64.

Compilando programas 32 bits

No caso de você precisar compilar um programa de 32 bits (wine e grub são dois exemplos de programas de código aberto que são apenas de 32 bits), primeiro configure o ambiente de shell do root executando o comando:

```
# . /etc/profile.d/32dev.sh
```

Observe o 'ponto' no início da linha - na verdade, é parte da linha de comando! O uso do ponto é equivalente ao comando 'source'.
A execução deste comando altera ou cria várias variáveis de ambiente. O efeito disso é que as versões de 32 bits dos binários são preferíveis aos binários de 64 bits quando você compila o código-fonte - você estará executando uma compilação de 32 bits. O efeito durará até que você saia do shell do root.

Neste ambiente alterado, você poderá usar o SlackBuilds padrão para construir pacotes de 32 bits para o Slackware64. Há algumas coisas a se ter em mente:

1. Você tem que definir a variável ARCH como 'i486' porque mesmo em seu computador 'x86_64' você está compilando um programa de 32 bits!
   Isto está relacionado ao *tripleto* de "$ ARCH-slackware-linux" que é normalmente usado no comando "configure".
   1. Como exceção, você terá que compilar o pacote "wine" com 'ARCH = x86_64' porque você instalará este pacote diretamente em seu computador multilib sem converter para um pacote 'compat32'.
2. Se você deseja instalar este pacote de 32 bits no Slackware64-multilib, você terá que convertê-lo em um pacote 'compat32':

```
# convertpkg-compat32 -i /path/to/your/fresh/foo-VERSION-i486-BUILD.tgz
# upgradepkg --install-new /tmp/foo-compat32-VERSION-x86_64-
BUILDcompat32.txz
```

Ressalvas

- Depois de instalar os pacotes "`-compat32`", você pode ter que reinstalar seus binários *Nvidia* ou *Ati* de drivers de vídeo para X.Org. Esses pacotes de driver contêm bibliotecas de 64 bits e 32 bits para serem o mais úteis possível em um sistema operacional multilib de 64 bits. Se você instalou os arquivos de driver para ambas as arquiteturas, o pacote "`mesa-compat32`" irá sobrescrever alguns dos arquivos de biblioteca de 32 bits.

  Por outro lado, se você originalmente *apenas* instalou o de 64 bits bibliotecas de driver para sua placa Nvidia/Ati, é recomendado após a instalação dos pacotes *multilib*, para reinstalar o pacote binário do driver. Desta vez, opte por instalar os arquivos de driver de 32 bits também.

  Os aplicativos gráficos de 32 bits que você irá executar em sua instalação multilib exigirão essas bibliotecas de driver de 32 bits. É provável que ocorram falhas se você não instalar os arquivos corretos.

- Se você deseja compilar seu kernel de 64 bits por conta própria, certifique-se de compilar a capacidade de emulação de 32 bits nele ou então o multilib irá falhar misteriosamente. Você precisará deste pedaço de configuração do kernel: **CONFIG_IA32_EMULATION**

Pacotes convertidos pelo massconvert32.sh

Esta é a lista de pacotes que são convertidos em versões "*-compat32*" pelo script `massconvert32.sh`. Observe que alguns desses pacotes não fazem parte do Slackware 13.0 ou 13.1, eles foram adicionados em uma versão posterior do Slackware, portanto, produzirão uma mensagem "*FALHA: pacote* '*nome_do_pacote*' *não foi encontrado!*" quando você execute o script em uma versão mais antiga. O contrário também é verdadeiro - alguns pacotes foram *removidos* em versões posteriores do Slackware e também irão disparar a mensagem "*FALHA: pacote* '*nome_do_pacote*' *não foi encontrado!*". Não se preocupe com isso.

```
# Série A/:

aaa_elflibs
attr
bzip2
cups
cxxlibs
dbus
e2fsprogs
eudev
libgudev
openssl-solibs
udev
```

```
util-linux
xz

# Série AP/:

cups
cups-filters
flac
mariadb
mpg123
mysql
sqlite

# Série D/:

libtool
llvm
opencl-headers

# Série L/:

SDL2
alsa-lib
alsa-oss
alsa-plugins
atk
audiofile
cairo
dbus-glib
elfutils
esound
expat
ffmpeg
fftw
freetype
fribidi
gamin
gc
gdk-pixbuf2
giflib
glib2
gmp
gnome-keyring
gtk+2
gst-plugins-base
gst-plugins-base0
gst-plugins-good
gst-plugins-good0
gst-plugins-libav
gstreamer
gstreamer0
```

```
hal
harfbuzz
icu4c
jasper
json-c
lame
lcms
lcms2
libaio
libart_lgpl
libasyncns
libclc
libedit
libelf
libexif
libffi
libglade
libgphoto2
libidn
libieee1284
libjpeg
libjpeg-turbo
libmng
libmpc
libnl3
libnotify
libogg
libpcap
libpng
libsamplerate
libsndfile
libtasn1
libtermcap
libtiff
libunistring
libusb
libvorbis
libxml2
libxslt
lzo
ncurses
ocl-icd
openjpeg
orc
pango
popt
pulseaudio
python-six
qt
readline
sbc
```

```
sdl
seamonkey-solibs
speexdsp
startup-notification
svgalib
v4l-utils
zlib

# Série N/:

curl
cyrus-sasl
gnutls
libgcrypt
libgpg-error
libtirpc
nettle
openldap-client
openssl
p11-kit
samba

# Série X/:

fontconfig
freeglut
glew
glu
libFS
libICE
libSM
libX11
libXScrnSaver
libXTrap
libXau
libXaw
libXcomposite
libXcursor
libXdamage
libXdmcp
libXevie
libXext
libXfixes
libXfont
libXfont2
libXfontcache
libXft
libXi
libXinerama
libXmu
libXp
```

```
libXpm
libXprintUtil
libXrandr
libXrender
libXres
libXt
libXtst
libXv
libXvMC
libXxf86dga
libXxf86misc
libXxf86vm
libdmx
libdrm
libepoxy
libfontenc
libinput
libpciaccess
libva
libva-intel-driver
libvdpau
libxcb
libxshmfence
mesa
pixman
vulkan-sdk
xcb-util

# Série XAP/:

sane
```

Mirrors para download do Multilib

Você pode baixar os pacotes multilib de (pelo menos) estes locais:

- http://slackware.com/~alien/multilib/
- http://bear.alienbase.nl/mirrors/people/alien/multilib/
- http://slackware.uk/people/alien/multilib/
- http://alien.slackbook.org/slackware/multilib/
- http://slackbuilds.org/mirror/alien/multilib/

Ferramentas de suporte de terceiros

- Sébastien Ballet escreveu uma ferramenta chamada *compat32pkg*. Em seu site ele tem *compat32pkg* disponível para download, bem como uma extensa documentação sobre como usá-lo no Slackware64.
  Vou citar o site:

"Compat32pkg é uma ferramenta automatizada que fornece todos os necessário para gerenciar (converter, instalar, atualizar, remover) a parte de 32 bits do multilib do AlienBob para slackware-64 e todos os pacotes de 32 bits do Slackware-32 para os quais os usuários possam encontrar necessidades em um ambiente de 64 bits, como firefox, seamonkey, jre, ... "

- Existe também slackpkg+, escrito por Matteo Rossini (apelidado de zerouno) com contribuições de (entre outros) Sèbastien Ballet. Este é um plugin para o próprio slackpkg do Slackware que adiciona a capacidade de instalar pacotes de repositórios externos (de terceiros) não oficiais do Slackware. Ele tem um bom suporte para adicionar multilib ao Slackware de 64 bits e mantê-lo atualizado.

Traduções

- Bruno Russo traduziu este artigo para o português (Brasil): http://www.brunorusso.eti.br/slackware/doku.php?id=multilib_para_o_slackware_x86_64
- Mehdi Esmaeelpour traduziu este artigo para o persa: http://www.slack-world.com/index.php/articles/43-general-system/85-multilib-slackware64
- Patrick FONIO e Sebastien BALLET traduziram este artigo para o francês: http://wiki.slackware-fr.org/avance:articles:slackware64-multilib

Agradecimentos

- Muito obrigado a Fred Emmott, que criou o Slamd64, o fork não oficial original do Slackware de 64 bits. Embora o Slackware64 não tenha sido baseado no trabalho de Fred, eu ainda aprendi muito do que sei sobre como configurar a parte de 32 bits de um Linux multilib com seus escritos que são encontrados no Slamd64.
  Observe que o Slamd64 tinha pacotes separados para 64 bits e 32 bits de gcc/glibc. No entanto, acredito que é mais limpo manter esses pacotes multilib essenciais não divididos. Eu segui o conceito já usado no próprio pacote *binutils* do Slackware64, que tem capacidade multilib de 64 bits e 32 bits agrupados em um pacote.

- Cross Linux From Scratch.
  O Wiki CLFS (http://trac.cross-lfs.org/wiki/read#ReadtheCrossLinuxFromScratchBookOnline) é uma 'leitura obrigatória' se você quiser entender como portar o Linux para uma nova arquitetura. Eu tirei várias idéias, conceitos e patches deles ao criar o Slackware64 do zero, e novamente quando criei meus pacotes multilib gcc/glibc do zero (meu README neste multilib-from-scratch está disponível no diretório ./source).

Divirta-se!

Eric

Fontes

- O artigo original, escrito por Eric Hameleers, está em http://alien.slackbook.org/dokuwiki/doku.php?id=slackware:multilib

slackware, multilib, author alienbob, translator carriunix