

Slackware ARM - Perguntas Frequentes

A maioria das perguntas são adicionadas aqui surgem de questões colocadas no [fórum do Slackware ARM](#).

Hardware

P: O ARM do Slackware será executado em <inserir nome do dispositivo>?

R: Esta é a pergunta mais frequente.

A resposta será “provavelmente”.

O Slackware ARM (todos os pacotes) será executado em qualquer cpu ARM que seja um armv5te ou superior. A maioria das CPUs atende a esses requisitos - especialmente novos dispositivos ARM.

É impossível que o Slackware ARM suporte cada dispositivo ARM pronto para uso. Ao contrário do mundo x86, onde um Kernel pode ser usado para todas as máquinas, no mundo ARM nós precisamos de um kernel por arquitetura ARM (daí porque o Slackware ARM tem dois pacotes de kernel - uma para a arquitetura System On Chip 'Kirkwood' e outra para a placa 'ARM Versatile'), Assim, a menos que o seu dispositivo seja atualmente suportado pelo Slackware ARM, você precisará compilar seu próprio kernel e determinar como ajustar o carregador de boot usado no dispositivo (a opção de inicialização é a mais comum). Não confunda os dados do usuário (os pacotes) com o Kernel. Não há necessidade de “portar” o Slackware ARM para outra arquitetura ARM (a menos que você queira recompilar alguns pacotes para melhor eficiência ao executar em uma CPU de nível superior à do ARMv5) - normalmente você só precisa de um novo kernel e dos módulos do kernel, e talvez alguns módulos de drivers adicionais para o X.org. Se você é novo no Linux e no ARM, isso representa uma enorme oportunidade de aprendizado para a qual você precisará reservar uma grande fatia de tempo, tenacidade, curiosidade e interesse!

Uma das maneiras mais fáceis de verificar se o seu dispositivo tem alguma semelhança com o suporte da comunidade Linux é verificar se o Debian ou o Fedora são executados nele. Se assim for, muitos dos usuários do Debian tendem a escrever uma excelente documentação descrevendo como eles instalaram o Linux em seus dispositivos. A maioria dessas informações pode ser usada para ajudar no encaixe do Slackware ARM no dispositivo.

P: Por que o Slackware ARM suporta o dispositivo X, mas não o Y?

R: Esta pergunta é frequentemente feita em referência ao Raspberry Pi, que de longe tem a mente mais ampla. O Slackware ARM é desenvolvido por [uma pessoa](#) cujo interesse em ARM é muito específico. Dispositivos ARM tem que ter Real Time Clocks (RTC), de preferência suporte SATA e mais importante ter suporte no kernel da linha principal. É impossível progredir um Sistema Operacional se você estiver preso a uma versão específica do Kernel (mesmo que seja por um curto período), pois isso atrasa o ciclo de desenvolvimento. Além disso, os dispositivos segmentados para suporte “oficial” (dispositivos suportados diretamente da árvore do ARM do Slackware) precisa ser aberto e não requer firmware propiciatório/licenciado e/ou alterado, a fim de permitir a inicialização de um sistema operacional alternativo. A comunidade do ARM do Slackware preenche lacunas onde há demanda.

Detalhes da instalação

P: Eu recebo um erro de CRC tentando inicializar o kernel

R: Você pode ter usado o modo de transferência ASCII se fez o download via FTP. Baixe novamente usando o modo de transferência BINÁRIA.

P: Eu estou tentando fazer o chroot no sistema de arquivos miniroot de uma instalação ARM Linux existente, mas recebo segfaults.

R: Isso geralmente ocorre porque o kernel que você está executando é mais antigo que o 2.6.31 (o glibc do Slackware ARM é compilado para requerer uma versão do kernel 2.6.31 ou superior).

Você precisa atualizar o kernel em sua instalação existente do ARM Linux.

Pacotes e espaço do usuário

P: O Slackware ARM tem um kernel mais novo do que o Slackware x86 - podemos esperar que o x86 acompanhe o problema?

R: O Slackware ARM tem como objetivo seguir o maior número possível de opções de configuração do kernel no x86, a fim de fornecer uma experiência de usuário consistente, mas o port para ARM usa qual um kernel que é o mais apropriado no momento em relação à arquitetura ARM (geralmente o kernel mais recente é o preferido, ou necessário para fornecer suporte para uma arquitetura recém-suportada).

Compilando software no ARM

Construindo os pacotes do Slackware ARM

P: Os scripts do SlackBuild no Slackware ARM não são executados. Como posso recriar os pacotes do Slackware ARM a partir do diretório de origem?

R: O arquivo [README](#) explica o processo.

P: Os pacotes do ARM do Slackware são compilados?

R: Não diretamente. Os pacotes são todos construídos nativamente em um sistema operacional do Slackware ARM, mas usam o distcc para um monte de máquinas x86/x86_64 que executam uma cadeia de ferramentas “cruzada” básica que combinam com as versões da toolchain e todas presentes no slackwarearm-current. O toolchain cruzado pode ser encontrada [aqui](#). Geralmente, ele foi testado e baseado no slackware64-current, mas não é suportado.

P: O Slackware ARM usa o Scratchbox?

R: O Slackware ARM usou o ScratchBox em 2002 para construir originalmente o primeiro lote de pacotes para inicializar o port, e algum esforço foi feito na produção de scripts para ajudar a

configurar um ambiente parecido com o Slackware. No entanto, este trabalho foi abandonado logo após o port ser capaz de se auto-sustentar.

Scripts externos do SlackBuild

P: Como mantenedor de scripts do SlackBuild, como posso diferenciar entre Slackware 14.2 (Software Floating Point) e Slackware 15.0 (Hardware Floating Point) ou posterior?

R: [Este trecho de código](#) verifica o quadlet de destino do gcc e define os sinalizadores padrão do compilador.

Software de Compilação Cruzada

P: Como posso cruzar o software de compilação para uso no Slackware ARM?

R: Se desejar, você pode cruzar o software de compilação usando um compilador cruzado adequado que produz binários do ARM. Existem vários compiladores cruzados disponíveis em formato binário, ou você pode construir o seu próprio (embora seja muito mais trabalhoso, mas você aprenderá muito!). A razão pela qual os pacotes do Slackware ARM são construídos nativamente é porque muitos pacotes não são adequados para compilação cruzada sem fazer muitos ajustes manuais para Makefiles e configurar scripts. Além disso, se o seu pacote estiver vinculado a bibliotecas do sistema, pode ser complicado fazer isso funcionar corretamente em um ambiente de compilador cruzado. Na minha experiência, é muito mais fácil obter uma máquina ARM rápida (um quad core Orange Pi) e usar o distcc. A quantidade de tempo que você gasta esperando pela conclusão da compilação está muito longe da quantidade de sistemas de criação de hacks que funcionam adequadamente em um ambiente "cruzado".

Geral

P: Por que o port do ARM foi renomeada de ARMedslack para Slackware ARM?

R: Originalmente, o port para ARM foi nomeada ARMedslack porque não era um port oficial e, como tal, não suportava o nome Slackware. Em 2009, o ARMedslack se tornou o port oficial, mas continuou com o nome ARMedslack até 2012, quando foi renomeado para Slackware ARM, a tempo para o lançamento do Slackware 14.0. O atraso na alteração do nome foi simplesmente devido à quantidade de trabalho necessária para fazê-lo!

[translated pt macgyverpt](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
<https://docs.slackware.com/pt-br:slackwarearm:faq>

Last update: **2019/08/29 17:53 (UTC)**

