



Guilherme Ferreira de Jesus 11811ETE008

Sistemas Embarcados
Buildroot e Yocto

11/05/2021

Sumário

Buildroot..... 3

 Vantagens..... 3

 Desvantagens 3

 Recomendação..... 4

Yocto..... 4

 Vantagens..... 4

 Desvantagens 5

 Recomendação..... 6

Resumindo..... 6

Buildroot

O projeto Buildroot é definido como "uma ferramenta simples, eficiente e fácil de usar para gerar sistemas Linux embarcados por meio de compilação cruzada". Ele compartilha muitos dos mesmos objetivos do projeto Yocto, no entanto, é focado na simplicidade e no minimalismo. Em geral, Buildroot desabilitará todas as configurações opcionais de tempo de compilação para todos os pacotes (com algumas exceções notáveis), resultando no menor sistema possível. Caberá ao designer do sistema habilitar as configurações apropriadas para um determinado dispositivo.

Buildroot cria todos os componentes da fonte, mas não oferece suporte ao gerenciamento de pacotes no destino. Como tal, às vezes é chamado de gerador de firmware, pois as imagens são amplamente corrigidas no momento da construção. Os aplicativos podem atualizar o sistema de arquivos de destino, mas não há mecanismo para instalar novos pacotes em um sistema em execução.

A saída do Buildroot consiste amplamente em três componentes:

- A imagem do sistema de arquivos raiz e quaisquer outros arquivos auxiliares necessários para implantar o Linux na plataforma de destino
- O kernel, boot-loader e módulos de kernel apropriados para o hardware alvo
- O conjunto de ferramentas usado para construir todos os binários de destino.

Vantagens

O foco do Buildroot na simplicidade significa que, em geral, é mais fácil de aprender do que Yocto. O sistema de construção principal é escrito em Make e é curto o suficiente para permitir que um desenvolvedor entenda todo o sistema enquanto é expansível o suficiente para atender às necessidades de desenvolvedores Linux embarcados. O núcleo Buildroot geralmente só lida com casos de uso comuns, mas é expansível por meio de scripts.

O sistema Buildroot usa Makefiles normais e a linguagem Kconfig para sua configuração. O Kconfig foi desenvolvido pela comunidade do kernel Linux e é amplamente usado em projetos de código aberto, tornando-o familiar para muitos desenvolvedores.

Devido ao objetivo do design de desabilitar todas as configurações opcionais de tempo de construção, Buildroot geralmente produzirá as menores imagens possíveis usando a configuração pronta para uso. Os tempos de construção e os recursos do host de construção também serão menores, em geral, do que os do projeto Yocto.

Desvantagens

O foco na simplicidade e nas opções de construção habilitadas mínimas implicam que você pode precisar fazer uma customização significativa para configurar uma construção Buildroot para seu aplicativo. Além disso, todas as opções de configuração são armazenadas em um único arquivo, o que significa que, se você tiver várias plataformas de hardware, precisará fazer cada uma das alterações de personalização para cada plataforma.

Qualquer mudança no arquivo de configuração do sistema requer uma reconstrução completa de todos os pacotes. Isso é um pouco atenuado pelos tamanhos mínimos de imagem e tempos de compilação em comparação com o Yocto, mas pode resultar em compilações longas enquanto você ajusta sua configuração.

O armazenamento em cache do estado do pacote intermediário não é habilitado por padrão e não é tão completo quanto a implementação do Yocto. Isso significa que, embora a primeira compilação possa ser mais curta do que uma compilação Yocto equivalente, as compilações subsequentes podem exigir a reconstrução de muitos componentes.

Recomendação

Usar Buildroot para seu próximo projeto de Linux embarcado é uma boa escolha para a maioria dos aplicativos. Se o seu projeto requer vários tipos de hardware ou outras diferenças, você pode querer reconsiderar devido à complexidade de sincronizar várias configurações, no entanto, para um sistema que consiste em uma única configuração, Buildroot provavelmente funcionará bem para você.

Yocto

O projeto Yocto é definido como "um projeto de colaboração de código aberto que fornece modelos, ferramentas e métodos para ajudá-lo a criar sistemas personalizados baseados em Linux para produtos embarcados, independentemente da arquitetura de hardware." É uma coleção de receitas, valores de configuração e dependências usadas para criar uma imagem de tempo de execução Linux customizada sob medida para suas necessidades específicas.

Divulgação completa: a maior parte do meu trabalho com Linux embarcado se concentrou no projeto Yocto, e meu conhecimento e preconceito para este sistema provavelmente serão evidentes.

Yocto usa Openembedded como seu sistema de construção. Tecnicamente, os dois são projetos separados; na prática, entretanto, os usuários não precisam entender a distinção e os nomes dos projetos são freqüentemente usados alternadamente.

A saída de uma construção de projeto Yocto consiste amplamente em três componentes:

- Binários de tempo de execução de destino: Incluem o carregador de inicialização, kernel, módulos do kernel, imagem do sistema de arquivos raiz. e quaisquer outros arquivos auxiliares necessários para implantar o Linux na plataforma de destino.
- Feed de pacote: esta é a coleção de pacotes de software disponíveis para serem instalados em seu destino. Você pode selecionar o formato do pacote (por exemplo, deb, rpm, ipk) com base em suas necessidades. Alguns deles podem ser pré-instalados nos binários de tempo de execução de destino, no entanto, é possível construir pacotes para instalação em um sistema implantado.
- Target SDK: trata-se da coleção de bibliotecas e arquivos de cabeçalho que representam o software instalado em seu destino. Eles são usados por desenvolvedores de aplicativos ao construir seu código para garantir que estejam vinculados às bibliotecas apropriadas

Vantagens

O projeto Yocto é amplamente utilizado na indústria e tem o apoio de muitas empresas influentes. Além disso, tem uma grande e vibrante comunidade de desenvolvedores e um ecossistema que contribuem para isso. A combinação de entusiastas do código aberto e patrocinadores corporativos ajuda a impulsionar o projeto Yocto.

Existem muitas opções para obter suporte com Yocto. Existem livros e outros materiais de treinamento, se você quiser fazer você mesmo. Muitos engenheiros com experiência em Yocto estão disponíveis se você quiser contratar especialistas. E muitas organizações comerciais fornecem produtos baseados em Yocto prontos para uso ou implementação e customização baseada em serviços para o seu projeto.

O projeto Yocto é facilmente expandido por meio de camadas, que podem ser publicadas independentemente para adicionar funcionalidade adicional, para plataformas de destino não disponíveis nas versões do projeto ou para armazenar personalizações exclusivas para seu sistema. Camadas podem ser adicionadas à sua configuração para adicionar recursos exclusivos que não estão especificamente incluídos nas versões de estoque; por exemplo, a camada "meta-navegador" contém receitas para navegadores da web, que podem ser facilmente construídas para o seu sistema. Como são mantidos de forma independente, as camadas podem estar em um cronograma de liberação diferente (ajustado para a velocidade de desenvolvimento das camadas) do que as versões padrão do Yocto.

Yocto tem indiscutivelmente o suporte de dispositivo mais amplo de qualquer uma das opções discutidas neste artigo. Devido ao suporte de muitos fabricantes de semicondutores e placas, é provável que o Yocto ofereça suporte a qualquer plataforma de destino que você escolher. Os lançamentos diretos do Yocto suportam apenas algumas placas (para permitir testes e ciclos de lançamento adequados), no entanto, um modelo de trabalho padrão é usar camadas de suporte de placa externas.

Finalmente, Yocto é extremamente flexível e personalizável. As personalizações para seu aplicativo específico podem ser armazenadas em uma camada para encapsulamento e isolamento. As personalizações exclusivas de uma camada de feição são geralmente armazenadas como parte da própria camada, o que permite que as mesmas configurações sejam aplicadas simultaneamente a várias configurações do sistema. Yocto também fornece uma prioridade de camada bem definida e capacidade de sobreposição. Isso permite definir a ordem em que as camadas são aplicadas e pesquisadas por metadados. Ele também permite que você substitua as configurações em camadas com prioridade mais alta; por exemplo, muitas personalizações para receitas existentes serão adicionadas em suas camadas privadas, com a ordem precisamente controlada pelas prioridades.

Desvantagens

A maior desvantagem do projeto Yocto é a curva de aprendizado. Leva muito tempo e esforço para aprender o sistema e realmente entendê-lo. Dependendo de suas necessidades, isso pode ser um investimento muito grande em tecnologias e competências que não são essenciais para sua aplicação. Nesses casos, trabalhar com um dos fornecedores comerciais pode ser uma boa opção.

Os tempos e recursos de build de desenvolvimento são bastante altos para builds de projetos Yocto. O número de pacotes que precisam ser construídos, incluindo o conjunto de ferramentas, kernel e todos os componentes de tempo de execução de destino, é significativo. As estações de trabalho de desenvolvimento para desenvolvedores Yocto tendem a ser grandes sistemas. Usar um notebook compacto não é recomendado. Isso pode ser atenuado usando servidores de compilação baseados em nuvem disponíveis de muitos provedores. Além disso, o Yocto possui um mecanismo de cache embutido que permite reutilizar componentes construídos anteriormente ao determinar que os parâmetros para construir um pacote específico não foram alterados.

Recomendação

Usar o projeto Yocto para seu próximo design de Linux embarcado é uma escolha forte. Das opções apresentadas aqui, é a mais amplamente aplicável, independentemente do seu caso de uso alvo. O amplo suporte da indústria, a comunidade ativa e o amplo suporte de plataforma tornam esta opção uma boa escolha para designers obrigatórios.

Resumindo

O Buildroot coloca todas as informações de configuração em um arquivo, que pode ser editado usando qualquer uma das interfaces da ferramenta kconfig do kernel (por exemplo, xconfig ou menuconfig). Esse arquivo especifica a arquitetura, o kernel, o carregador de inicialização e os pacotes de espaço do usuário a serem incluídos. A desvantagem é que máquinas semelhantes não podem compartilhar nenhuma informação de configuração: deve-se construir cada destino separadamente.

Já o Yocto, mantém as informações de configuração em várias partes. A configuração da distribuição dita pacotes gerais, conjuntos de ferramentas e bibliotecas; a configuração da máquina descreve a arquitetura e os recursos de hardware; a receita da imagem lista quais pacotes devem ser pré-instalados; a configuração local controla como a construção é executada, desse modo, pode ser necessário reunir muitas camadas antes de iniciar uma construção, mas também torna mais fácil construir variações com pouco trabalho adicional.