

WINDRIVER
SUPPORT NETWORK

Wind River Linux Distro Quick Start

WIND RIVER LINUX DISTRO QUICK START



著作権について

Copyright © 2023 Wind River Systems, Inc.

無断転載を禁じます。この出版物のいかなる部分も、Wind River Systems, Incの書面による事前の許可なしに、いかなる形式または手段によっても複製または配布することはできません。

Wind River、Simics、VxWorksはWind River Systems, Incの登録商標です。Wind Riverのロゴは、Wind River Systems, Incの商標です。記載されているサードパーティの商標は、それぞれの所有者に帰属します。Wind Riverの商標に関する詳細は、以下をご参照ください。

www.windriver.com/company/terms/trademark.html

本製品には、サードパーティからウインドリバーにライセンスされたソフトウェアが含まれている場合があります。ウインドリバーのダウンロードおよびインストールポータル「Wind River Delivers」には、製品に関連する通知が必要に応じて掲載されています。

<https://gallery.windriver.com/>

<https://windshare.usa.windriver.com/>

ウインドリバーは、情報提供を目的として、出版物を掲載したり、第三者のウェブサイトへのリンクを提供することで、第三者の文書を参照することがあります。ウインドリバーは、このような第三者のドキュメントに記載されている情報について一切の責任を負いません。

Wind River

500 Wind River Way

Alameda, CA 94501-1153

U.S.A.

Toll free (U.S.A.): +1-800-545-WIND

Telephone: +1-510-748-4100

Facsimile: +1-510-749-2010

その他の連絡先については、下記のウェブサイトをご覧ください。

www.windriver.com

カスタマーサポートへのお問い合わせ

www.windriver.com/support

※ 本ドキュメントは、参照目的のために英語版「WIND RIVER LINUX DISTRO QUICK START」を翻訳したものです。

目次

1. クイックスタートの概要	1
2. Wind River Linux Distroのダウンロード	3
3. ターゲットシステムイメージ クイックスタート	4
3.1. Ignition Configurationの作成	5
3.2. ハードウェア上でターゲットシステムイメージの起動	5
3.3. QEMUによるターゲットシステムイメージの起動	5
4. パッケージアップデートのためのDistroイメージの作成	8
5. リリース情報	10
6. 追加情報の入手先	11

1. クイックスタートの概要

このクイックスタートの説明を参考にして、Wind River® Linux Distroをご利用ください。

Wind River Linux Distroは、あらかじめ定義されたディストリビューションと、バイナリイメージやパッケージの利便性を求めるユーザ向けのバイナリLinuxディストリビューションです。2つのオープンソースプロジェクト、CoreOS ProjectとYocto Projectのテクノロジーを取り入れています。

Yocto ProjectベースのWind River Linux LTS23は、ソフトウェアパッケージとアプリケーションのSDKのビルドに使用されています。現在は**x86-64**に対応しており、今後は**Aarch64**と**Risc-V**にも対応し、各アーキテクチャに1つのユーザー空間を提供できる予定です。

CoreOSプロジェクトのツールは、イメージへのパッケージアセンブリ、ターゲットへのイメージのインストール、ターゲットにインストールされたソフトウェアのカスタマイズや、ターゲットのアップデートに使用されています。

CoreOSはFedoraプロジェクトの一環です。CoreOSのツールは、ユーザのインフラストラクチャを保護、簡素化、自動化するために使用されます。

Wind River Linux Distroで使用されているツール：

- **rpm-ostree** - ハイブリッドイメージ/パッケージシステムです。**libostree**と**libdnf**を組み合わせたもので、ostreeイメージ、または、パッケージ管理アプローチを使用して、システムの拡張やアップグレードを可能にします。
- **Ignition** - インストール時および初回起動時にディスクと**initramfs**を操作する低レベルツールです。様々な方法でIgnitionコンフィギュレーションが適用できます。
- **butane** - Butaneのコンフィギュレーションを、機械が読み取れるIgnition configsに変換します。Butaneは通常コンテナとして提供されます。
- **bootupd** - ディストリビューションに依存しないブートローダのアップデートを行います。
- **coreos-installer** - CoreOSのインストーラです。通常は、コンテナ内またはライブイメージから実行されます。インストールを完了するだけでなく、以下も実行できます。
 - OSイメージのダウンロードと検証
 - ダウンロード可能なイメージのリストアップ
 - ライブISOイメージにIgnition configを埋め込み、そこから起動する実行システムのカスタマイズ
 - Ignition configを**initrd**イメージにラップし、ライブPXE **initramfs**に追加することで、そこから起動する実行システムのカスタマイズ

このドキュメントでは、Wind River Linux Distroでデバイスまたは仮想マシンを起動する手順をご紹介します。Wind River Linux Distroターゲットシステムまたは仮想マシンが起動したら、オプションでSDKをインストールしてアプリケーション開発に使用することができます。

必要条件

- Dockerエンジンがインストールされ、権限のないユーザがDockerコマンドを実行できるように設定されたLinuxホスト
- 仮想マシンでDistroを実行する場合、最低16GBのRAMとQEMUのインストール可能であれば権限のないユーザが**/dev/kvm**にアクセス可能
- イメージのダウンロードや、システムのアップデートを行うためのインターネットアクセス
- Linuxおよびコマンドラインに関する中級レベル知識
これには、コマンドの実行、パッケージのインストール、プロジェクトの設定ファイルの編集などが含まれます。

ワークフロー

Wind River Linux Distroのバイナリイメージを稼働させるためのワークフローは以下の通りです。

- イメージファイルをダウンロードします。
詳細については、「[Wind River Linux Distroのダウンロード \(P.3\)](#)」を参照してください。
- Ignition設定ファイルを作成します。詳細については、「[Ignition Configurationの作成 \(P.4\)](#)」を参照してください。
- ハードウェア、またはQEMU上でWind River Linux Distroイメージを起動します。
 - [ハードウェア上でターゲットシステムイメージの起動 \(P.4\)](#)
 - [QEMUによるターゲットシステムイメージの起動 \(P.5\)](#)
 - DNF Package Managerを使用して、アップデート用のイメージを作成します。詳細については、「[パッケージのアップデートのためのDistroイメージ作成 \(P.8\)](#)」参照してください。

2. Wind River Linux Distroのダウンロード

Wind River coreos-installerコンテナを使って、Wind River Linux Distroイメージをダウンロードすることができます。

手順

Wind River coreos-installerコンテナからWind River Linux Distroをダウンロードします。

```
$ docker run --rm -v $(pwd):/data -w /data docker.io/windriver/wrdistro23:coreos-installer  
download
```

3. ターゲットシステムイメージ クイックスタート

3.1. Ignition Configurationの作成

Ignition configはJSONファイルとしてフォーマットされます。

本作業について

Ignition configのプロセスには、Butane configを作成するステップと、Butane configをJSON Ignition configに変換するステップの2つがあります。

手順

1. パスワードをハッシュ化します。

```
$ docker run -ti --rm quay.io/coreos/mkpasswd --method=yescrypt YOUR_PASSWORD > password_hash.txt
```

2. Butane configを作成します。

```
$
cat <<-EOF sample.bu
variant: fcos
version: 1.4.0
passwd:
  users:
    - name: core
      password_hash: $(cat password_hash.txt)
kernel_arguments:
  should_exist:
    - console=ttyUSB0,115200n8
  should_not_exist:
    - console=tty0
    - console=ttyS0,115200n8
EOF
```

3. Butaneコンテナを起動して、Butane configをIgnition configに変換します。

```
$ docker run --rm -v $(pwd)/sample.bu:/config.bu:z \ docker.io/windriver/wrdistro23:butane --pretty --strict /config.bu > ./sample.ign
```

3.2. ハードウェア上でターゲットシステムイメージの起動

ターゲットシステムイメージをハードウェアデバイスのストレージにコピーしたら、そのイメージを起動して使い始めることができます。

本作業について

USBフラッシュドライブがデバイス/**dev/sdX**を取ると仮定して、coreos-installerを使ってイメージをコピーします。次に、USBデバイスをターゲットのブート可能なUSBソケットに挿入し、電源を入れます。尚、システムがGrubブートメニューで起動している必要があります。

手順

1. Ignition configファイルをUSBフラッシュドライブに書き込む前に、USBフラッシュドライブのマウントを解除します。

マウントされているすべてのパーティションを特定します。例えば、USBデバイス/**dev/sdX**の場合：

```
$ mount | grep sdX
```

```
$ umount /dev/sdX1
```

```
$ umount /dev/sdX2
```

注：Linuxのディストリビューションには、USBドライブが接続されると自動的にマウントするものがあります。

2. Ignition configファイルをUSBフラッシュドライブに書き込みます。
ホスト上のファイルにアクセスするには、**/dev**、**/run/udev**、および **data**ディレクトリにbind-mountする必要があります。
例えば、Wind River coreos-installerコンテナを以下のように使用します。

```
$ sudo docker run --privileged --rm -v /dev:/dev -v /run/udev:/run/udev -v $(pwd):/data -w /data docker.io/windriver/wrdistro23:coreos-installer install /dev/sdX --ignition-file ./sample.ign --insecure-ignition
```

これでブート可能なUSBフラッシュ・デバイスが作成されます。

3. USBフラッシュ・デバイスをターゲットのブート可能なUSBソケットに差し込み、電源を入れます。
システムはGrubブートメニューで起動します。

注：ライブイメージからcoreos-installerを実行して、イメージを他のディスクにインストールすることが可能になります。

イメージが起動したら、DNF Package Managerを使ってパッケージのアップデートの準備をする必要があります。これにより、パッケージのインストール、アップデート、削除ができるようになります。詳細については、「[パッケージのアップデートのためのDistroイメージ作成 \(P.8\)](#)」を参照してください。

3.3. QEMUによるターゲットシステムイメージの起動

ターゲットシステムイメージをLinuxホストシステムに展開すると、一時的なストレージを作成してQEMUでイメージを起動することができます。

始める前に

- お使いのLinuxホストシステムには、`/usr/bin/qemu-system-arch` のバイナリ（最低バージョン2.11）がインストールされているかをご確認ください。
- お使いのLinuxホストシステムにWind River Linux Distroターゲットシステムイメージをダウンロードして展開してください。詳細については、「[Wind River Linux Distroのダウンロード（P.3）](#)」を参照してください。
- インターネットに接続しているかをご確認ください。

手順

- 16GBのディスクイメージを作成します。

```
$ qemu-img create -f raw boot-image-qemu.hddimg 16G
```

- OVMFイメージを作成します。

```
$ wget https://distro.windriver.com/dist/distro23/x86-64/OVMF_CODE.fd
```

```
$ wget https://distro.windriver.com/dist/distro23/x86-64/OVMF_VARS.fd
```

- ダウンロードしたイメージを、前のステップで作成した16GBのディスクイメージにコピーします。

```
$ xzcat wr-distro-*-metal.x86_64.raw.xz | dd of=boot-image-qemu.hddimg conv=notrunc
```

- 前のステップでコピーしたイメージをデプロイします。

次の表のコマンド例では、Linuxホスト・システム上のQEMUバイナリを参照しています。

BSP	実行するQEMUコマンド
intel-x86-64	オプション1 - KVMあり（推奨） <pre>\$ qemu-system-x86_64 -net nic -net user -m 4096 \ -cpu host -enable-kvm \ -fw_cfg name=opt/com.coreos/config,file=./sample.ign \ -drive if=pflash,format=qcow2,file=OVMF_CODE.fd \ -drive if=pflash,format=qcow2,file=OVMF_VARS.fd \ -drive if=none,id=hd,file=boot-image-qemu.hddimg,format=raw \ -device virtio-scsi-pci,id=scsi \ -device scsi-hd,drive=hd \ -nographic</pre> この例では、kvmオプションを使用してブート時間を短縮し、ターゲットの応答性を高めています。このオプションを使用するには、Linuxホストシステムの <code>/dev/kvm</code> へのアクセス権が必要です。詳細については、https://wiki.yoctoproject.org/wiki/How_to_enable_KVM_for_Poky_qemu を参照してください。 オプション2 - KVMなし

BSP	実行するQEMUコマンド
	<pre>\$ qemu-system-x86_64 -net nic -net user -m 4096 \ -cpu Nehalem \ -fw_cfg name=opt/com.coreos/config,file=./sample.ign \ -drive if=pflash,format=qcow2,file=OVMF_CODE.fd \ -drive if=pflash,format=qcow2,file=OVMF_VARS.fd \ -drive if=none,id=hd,file=boot-image-qemu.hddimg,format=raw \ -device virtio-scsi-pci,id=scsi \ -device scsi-hd,drive=hd \ -nographic</pre>

QEMUのコマンドとその意味についての詳細は、オンラインのREADMEファイルをご参照ください。

- **target_intel-x86-64.README.md** : <https://distro.windriver.com/dist/distro23/x86-64/x86-64.README.md>

イメージが起動したら、DNF Package Managerを使ってパッケージのアップデートの準備をする必要があります。これにより、パッケージのインストール、アップデート、削除ができるようになります。詳細については、「[パッケージのアップデートのためのDistroイメージ作成（P.8）](#)」を参照してください。

4. パッケージのアップデートのためのDistroイメージ作成

Wind River Linux Distroのバイナリイメージを使用する利点の一つは、アップデートを行う機能が各イメージにデフォルトで含まれていることです。

本作業について

Wind River Linux Distroイメージは、DNFパッケージマネージャによるオンターゲット・パッケージ・アップデートをサポートしています。これにより、DNFを使用してWind River Linux Distroリポジトリからパッケージをインストールすることができます。

始める前に

Wind River Linux Distroバイナリイメージでアップデートを実行するには、ターゲットシステムイメージを起動する必要があります。詳細については、「[ハードウェア上でターゲットシステムイメージの起動 \(P.4\)](#)」, 「[QEMUによるターゲットシステムイメージの起動 \(P.5\)](#)」をご参照ください。

手順

1. Wind River Linux Distro のバイナリイメージが起動し、rootとしてログインしていることを確認します。

2. イメージのロックを解除します。

デフォルトでは、イメージはロックされており、いかなるアップデートも許可されていません。パッケージのアップデートを成功させるためには、以下のコマンドを実行する必要があります。

```
$ ostree admin unlock --hotfix
```

3. groupとpasswdを手動でマージします。

このステップは、DNFが正しく動作するために必要です。デフォルトでは、rpm-ostreeは/etc/passwdを/usr/lib/passwdに、/etc/groupを/usr/lib/groupに分割し、NSSモジュールを使用して、代わりのロケーションである/usr/libに格納されている/etc/passwdと/etc/groupと同じ形式のファイルからユーザー情報を読み込みます。詳細は、<https://github.com/coreos/rpm-ostree/blob/main/rust/src/passwd.rs#L107>と<https://github.com/aperezdc/nss-altfiles#building>を参照ください。

useraddコマンドとgroupaddコマンドは関連するファイルの場所が決まっているため、groupaddコマンドやuseraddコマンドを必要とするパッケージでは、パッケージのプリインストールスクリプトが失敗する可能性があります。これを防ぐには、/usr/lib/passwdを/etc/passwd、/usr/lib/passwdを/etc/passwdに、/usr/lib/groupを/etc/groupに手動でマージし、重複を避けるためにusr/lib/passwdとusr/lib/groupを空にする必要があります。

以下のコマンドを実行してgroupとpasswdをマージし、DNFを正しく使用できるようにします。

```
$ cat /usr/lib/group >> /etc/group
$ echo "" > /usr/lib/group

$ cat /usr/lib/passwd >> /etc/passwd
$ echo "" > /usr/lib/passwd
```

4. DNFでパッケージのインストール、アップデート、削除を行います。

注意：DNFによるパッケージアップデートは、システムイメージとコンテナイメージの両方でサポートされています。

更新タイプ	実行するコマンド
インストール	<pre>\$ dnf install packageName</pre>
削除	<pre>\$ dnf remove packageName</pre>
パッケージインデックスの更新	DNFパッケージマネージャーは、パッケージのフィードインデックスをローカルにキャッシュします。このコマンドは、新しいフィードインデックスを読み込むために必要です。パッケージ管理ツールはこれらのインデックスを定期的に更新しますが、パッケージフィードの変更をすぐに確認するためには、明示的にキャッシュを更新する必要があります。 <pre>\$ dnf update</pre>
パッケージをアップグレードする	<pre>\$ dnf upgrade packageName</pre>

5. リリース情報

Wind River Linux Distroイメージの利用可能な機能とサポート情報を確認します。

パッケージ情報

各イメージには、デフォルトでrpm-ostreeによるパッケージ管理機能が搭載されており、パッケージの追加、更新、削除を行うことができるほか、追加のパッケージフィードも含まれています。

ハードウェア機能

サポートされているボード

- Intel NUC Platform: KabyLake-U Processor Sunrise Point-LP PCH, (NUC7i5DNK1E)
- qemu86-64

注： その他の機器固有の情報については、機器メーカーのドキュメントを参照してください。

対応する外部メモリデバイス

メモリデバイスの種類	詳細
USBフラッシュドライブ	Wind River Linux Distroのインストールには、通常16GB以上のフラッシュドライブを使用します。 イメージを保存するのに十分な容量と、アプリケーションの追加やシステムログファイルの保存、システムのアップデートによる拡張に備えて、十分な容量があるものを選びましょう。

6. 追加情報の入手先

Wind River Linuxプラットフォームのプロジェクトイメージが完成したので、さらに開発を進めたいと思うかもしれません。詳細については、以下をご参照ください。

Wind River Linux のドキュメント

すべてのドキュメントは、<https://docs.windriver.com> からオンラインで入手できます。

All Products > Wind River Linuxを選択し、「Wind River Linux LTS 23」を選択します。

サポートされているBSP、ワークフロー、製品のアップデートなど、一般的な製品情報については、[Wind River Linux Getting Started](#)をご参照ください。

Wind River Linux の RSSフィードを購読するには、お使いの RSSリーダーに https://support2.windriver.com/feeds/wrlinux_lts.xml 指定してください。

外部ドキュメント

Wind River Linuxは、OpenEmbedded Core Projectをベースにした Yocto Projectと同じビルドシステムを共有しています。そのため、Yocto Projectのドキュメントを参考にして、プラットフォーム・プロジェクト・イメージをさらに発展させることができます。

まずは、Yocto Projectビルドシステムを使ってプラットフォームプロジェクトのビルドを開発する際に使われる用語を理解することから始めましょう。

詳しくは、[Yocto Project MegaManual](#) : [Yocto Project Terms](#)をご覧ください。

外部の情報源	URL
The Yocto Project	Online: http://www.yoctoproject.org
Core OS	Online: https://github.com/coreos
OpenEmbedded Core (OE-Core)	Online: http://www.openembedded.org/wiki/OpenEmbedded-Core
QEMU	Online: http://wiki.qemu.org