



Red Hat Enterprise Linux 7

カーネルクラッシュダンプガイド

カーネルクラッシュダンプの設定と分析

Jaromír Hradílek

Petr Bokoč

カーネルクラッシュダンプの設定と分析

Jaromír Hradílek
Red Hat Customer Content Services
jhradilek@redhat.com

Petr Bokoč
Red Hat Customer Content Services
pbokoc@redhat.com

法律上の通知

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

カーネルクラッシュダンプガイドでは Red Hat Enterprise Linux 7 での kdump クラッシュリカバリーサービスの設定方法、検証方法、使い方などについて説明している他、crash デバッグユーティリティで生成したコアダンプを分析する方法についても簡単に触れています。本ガイドは Red Hat Enterprise Linux に関して基本的な知識をお持ちのシステム管理者を対象としています。

目次

第1章 kdump について	2
1.1. kdump と kexec について	2
1.2. メモリーの要件	2
第2章 kdump のインストールと設定	3
2.1. kdump のインストール	3
2.2. コマンドラインで kdump を設定する	4
2.3. グラフィカルユーザーインターフェースで kdump を設定する	8
2.4. kdump 設定のテスト	13
2.5. その他のリソース	13
第3章 コアダンプの分析	15
3.1. crash ユーティリティのインストール	15
3.2. crash ユーティリティの実行	15
3.3. メッセージバッファの表示	16
3.4. バックトレースの表示	17
3.5. プロセスの状態表示	18
3.6. 仮想メモリ情報の表示	18
3.7. オープンファイルの表示	19
3.8. ユーティリティの終了	19
付録A よくある質問	20
付録B kdump で対応している設定とダンプ出力先	21
B.1. kdump メモリー要件	21
B.2. メモリー自動予約の最小しきい値	21
B.3. 対応している kdump のダンプ出力先	22
B.4. 対応している kdump のフィルターレベル	22
B.5. 対応しているデフォルトの動作	23
付録C 改訂履歴	24

第1章 kdump について

1.1. kdump と kexec について

Kdump とは、システムのメモリ内容を保存して後で分析できるようカーネルのクラッシュをダンプするメカニズムを指します。**kexec** に依存します。**kexec** は別のカーネルのコンテキストから Linux カーネルを起動、BIOS を迂回して通常は失われてしまう 1 番目のカーネルメモリーの内容を維持することができます。

システムクラッシュが発生すると kdump は kexec を使って 2 番目のカーネルで起動します (キャプチャーカーネル)。この 2 番目のカーネルはシステムメモリーの予約部分に収納されていて 1 番目のカーネルからはアクセスできません。2 番目のカーネルは起動するとクラッシュしたカーネルメモリーの内容 (クラッシュダンプ) をキャプチャーして保存します。

1.2. メモリーの要件

kdump でカーネルクラッシュのダンプをキャプチャーして分析のため保存するにはキャプチャーカーネル用にシステムメモリーの一部を永続的に予約しておく必要があります。予約するとその部分はメインカーネルでは使用できなくなります。

メモリーの要件はシステムの構成要素によって異なります。そのひとつとしてシステムのハードウェアアーキテクチャーが挙げられます。次のコマンドをシェルプロンプトで入力してマシンの正確なアーキテクチャー名を検索 (**x86_64** など)、標準出力に表示させます。

```
uname -m
```

予約すべきメモリーサイズを左右する別の要因として搭載しているシステムメモリーの総量が影響します。例えば、x86_64 アーキテクチャーでは予約メモリーは 4 KB の RAM ごと 160 MB + 2 ビットになります。搭載されている物理メモリーの合計が 1 TB のシステムの場合には 224 MB ということになります (160 MB + 64 MB)。システムアーキテクチャーごとの kdump メモリー要件と物理メモリー量についての詳細は「[kdump メモリー要件](#)」を参照してください。

ほとんどのシステムでは必要なメモリー量は kdump によって自動的に算出、予約が行われます。この動作はデフォルトで有効になっていますが特定サイズ以上のメモリーが搭載されているシステムに限られます。この自動割り当て動作に必要なメモリーサイズはシステムのアーキテクチャーによって異なります。「[メモリー自動予約の最小しきい値](#)」に自動メモリー割り当てに必要な最小メモリーサイズの一覧をシステムアーキテクチャーごとに示します。

システムメモリーが自動割り当ての動作に必要な最小メモリーに満たない場合、または独自の予約メモリーサイズを必要とするような場合には予約メモリーを手作業で設定することができます。コマンドラインでこの作業を行う場合は「[メモリー使用を設定する](#)」を参照してください。グラフィカルユーザーインターフェースでこの作業を行う場合は「[メモリー使用を設定する](#)」を参照してください。



重要

kdump サービスを設定したら、自動メモリー割り当てであっても設定のテストを行うことを強く推奨します。設定のテスト方法については「[kdump 設定のテスト](#)」を参照してください。

第2章 kdump のインストールと設定

2.1. kdump のインストール

多くの場合、新しい Red Hat Enterprise Linux 7 のインストールで **kdump** サービスはデフォルトでインストールされます。グラフィカルインターフェースまたはテキストインターフェースを使って対話式のインストールを行っている場合、**Anaconda** インストーラーに kdump の設定画面があります。インストーラーの画面には **Kdump** というタイトルが付けられ、メイン画面の**インストールの概要**からアクセスすることができます。行える設定については制限があり、ここでは kdump を有効にするかどうかと予約するメモリーサイズを選択を行うことができます。kdump のメモリー要件については「[kdump メモリー要件](#)」を参照してください。インストーラーの kdump 設定画面については「[Red Hat Enterprise Linux 7 インストールガイド](#)」で説明しています。

注記

Red Hat Enterprise Linux の旧リリースでは **Firstboot** ユーティリティーで kdump の設定を行うことができました。このユーティリティーは **インストールの終了後**、システムをはじめて再起動すると自動的に実行されていました。Red Hat Enterprise Linux 7.1 からは kdump の設定がインストーラー内に移動しています。

カスタムのキックスタートを使ったインストールなど一部のインストール方法ではデフォルトで kdump をインストールしないまたは有効にしない場合があります。このような場合、kdump を追加でインストールするには **root** で次のコマンドをシェルプロンプトから実行します。

```
# yum install kexec-tools
```

kdump とその他必要なパッケージがすべてインストールされます。システムにアクティブなサブスクリプションがある、またはシステムのアーキテクチャーに応じた **kexec-tools** パッケージが収納されたカスタムのリポジトリがあると前提されています。

注記

システムに kdump がインストールされているかわからない場合は **rpm** を使うと確認できます。

```
$ rpm -q kexec-tools
```

この他、グラフィカルな設定ツールもあります。ただし上記のコマンドを使った場合、デフォルトではグラフィカルな設定ツールはインストールされません。グラフィカルな設定ツールをインストールする場合は別のコマンドを **root** で実行します。グラフィカルな設定ツールの詳細は「[グラフィカルユーザーインターフェースで kdump を設定する](#)」を参照してください。

```
# yum install system-config-kdump
```

Yum パッケージマネージャーを使って Red Hat Enterprise Linux 7 に新しいパッケージをインストールする方法については「[Red Hat Enterprise Linux 7 システム管理者のガイド](#)」を参照してください。



重要

現在実装されている **Intel IOMMU** ドライバーの制限により **kdump** サービスでコアダンプイメージのキャプチャーが行えないことがあります。kdump を Intel アーキテクチャーで確実に使用するため IOMMU サポートは無効にすることをお勧めします。

2.2. コマンドラインで **kdump** を設定する

2.2.1. メモリー使用を設定する

kdump カーネル用に予約されるメモリーは必ずシステムの起動時にその予約が行われます。つまり、メモリーのサイズはシステムのブートローダー設定で指定されています。このセクションでは **GRUB2** ブートローダーを使用している AMD64 および Intel 64 のシステムでの予約メモリーサイズの変更方法、および **zipl** を使用している IBM Power System のサーバーでの予約メモリーサイズの変更方法について説明します。

手順2.1 **GRUB2** でメモリーオプションを変更する

1. **root** になり、**vim** や **Gedit** などのプレーンテキストエディターを使って `/etc/default/grub` 設定ファイルを開きます。
2. **GRUB_CMDLINE_LINUX** で始まる行を探します。

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto  
rd.lvm.lv=rhel/root rhgb quiet"
```

強調表示している **crashkernel=** オプションの部分で予約メモリーの設定を行います。

3. **crashkernel=** オプションの値を設定したいメモリーサイズに変更します。たとえば、128 MB に設定する場合は次のように変更します。

```
crashkernel=128M
```



注記

起動時にシステムで使用できる RAM のサイズに応じてオフセットや複数のメモリーサイズを指定するなど、予約メモリーの設定方法は数種類あります。このセクションではこうした設定方法について説明しています。

ファイルを保存してからエディターを終了します。

4. 変更を行った **default** ファイルを使って **GRUB2** 設定を再生成します。BIOS ファームウェアを使用しているシステムなら次のコマンドを実行します。

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

UEFI ファームウェアを使用しているシステムの場合は次のコマンドを実行します。

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```


上記の手順を完了するとブートローダーが再構成され、次回の再起動から設定ファイルに指定したメモリーサイズが予約されるようになります。

手順2.2 zipl でメモリーオプションを変更する

1. **root** になり、**vim** や **Gedit** などのプレーンテキストエディターを使って `/etc/zipl.conf` 設定ファイルを開きます。
2. **parameters=** セクションを探し、**crashkernel=** パラメーターを編集します (このパラメーターがない場合は追加する)。たとえば 128 MB のメモリー予約の場合は次のように編集します。

```
crashkernel=128M
```

注記

起動時にシステムで使用できる RAM のサイズに応じてオフセットや複数のメモリーサイズを指定するなど、予約メモリーの設定方法は数種類あります。このセクションではこうした設定方法について説明しています。

ファイルを保存してからエディターを終了します。

3. **zipl** 設定を再生成します。

```
# zipl
```

注記

zipl コマンドにオプションを何も付けずに実行するとデフォルト値が使用されます。使用できるオプションについては **zipl (8)** の man ページをご覧ください。

上記の手順を完了するとブートローダーが再構成され、次回の再起動から設定ファイルに指定したメモリーサイズが予約されるようになります。

crashkernel= オプションの指定方法は数種類あります。**auto** の値を使用すると「[kdump メモリー要件](#)」に記載したガイドラインに沿ってシステムのメモリー総合計に基づいた予約メモリーが自動設定されます。**auto** 値を特定のメモリーサイズに置き換えるとその動作が変更されます。たとえば、128 MB のメモリーを予約させる場合は次のように指定します。

```
crashkernel=128M
```

搭載しているメモリーの合計サイズに応じて予約メモリーのサイズが変化するように設定することもできます。可変のメモリー予約を設定する場合の構文は **crashkernel=<range1>: <size1>, <range2>: <size2>** になります。例を示します。

```
crashkernel=512M-2G:64M,2G-:128M
```

上記の例の場合、システムメモリーの合計サイズが 512 MB 以上 2 GB 未満の場合は 64 MB のメモリー、システムメモリーの合計サイズが 2 GB 以上の場合は 128 MB のメモリーが kdump 用に予約されます。

一部のシステムでは特定の固定オフセットをつけてメモリーの予約を行う必要があるシステムがあります。オフセットを設定すると予約メモリーはそこから開始されます。予約メモリーにオフセットを設ける場合に次の構文を使用します。

```
crashkernel=128M@16M
```

上記の例の場合、`kdump` は 128 MB のメモリー予約を 16 MB (物理アドレス 0x01000000) から開始することになります。オフセットパラメーターを 0 に設定する、または完全に省略すると `kdump` により自動的にオフセットが設けられます。前述の可変の予約メモリーを設定する場合にもこの構文を使用することができます。オフセット指定は必ず一番後ろに付けます (例、**`crashkernel=512M-2G:64M,2G-:128M@16M`**)。

2.2.2. `kdump` タイプを設定する

カーネルクラッシュがキャプチャーされたら、そのコアダンプをファイルとしてローカルのファイルシステムに保存するか、デバイスに直接書き込み **NFS** (Network File System) や **SSH** (Secure Shell) などのプロコルを使ってネットワーク経由で送信することができます。現在、設定できるのはいずれかひとつのオプションのみで、デフォルトのオプションは **vmcore** ファイルをローカルファイルシステムの `/var/crash/` ディレクトリーに保存します。このオプションを変更する場合は **root** になりテキストエディターで `/etc/kdump.conf` 設定ファイルを開き以下の説明に従ってオプションを変更します。

コアダンプの保存先のローカルディレクトリーを変更する場合は **#path /var/crash** 行の先頭にあるハッシュ記号 (「#」) を取り除き、値を変更先のディレクトリーパスに置き換えます。

```
path /usr/local/cores
```



重要

Red Hat Enterprise Linux 7 では `kdump` のダンプ出力先として **path** ディレクティブで指定されているディレクトリーが `kdump systemd` サービスの起動時に存在していなければなりません。起動時にこのディレクトリーが存在しないとサービスの起動は失敗します。この動作は Red Hat Enterprise Linux の旧リリースと異なります。旧リリースではサービスの起動時にこのディレクトリーが存在しないと自動的に作成されていました。

また、ファイルを別のパーティションに書き込みたい場合は **#ext4** 行の先頭のハッシュ記号 (「#」) を取り除き、値を変更先のディレクトリーパスに置き換えます。値にはデバイス名 (**#ext4 /dev/vg/lv_kdump** 行)、ファイルシステムのラベル (**#ext4 LABEL=/boot** 行)、UUID (**#ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937** 行) のいずれかを使用できます。例を示します。

```
ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937
```



重要

ストレージデバイスの指定は **LABEL=** または **UUID=** を使用することをお勧めします。`/dev/sda3` などディスクデバイス名は再起動を行うと同じ名前が使用されない恐れがあります。永続的なディスクデバイスの命名については [『Red Hat Enterprise Linux 7 ストレージ管理ガイド』](#) を参照してください。

ダンプをデバイスに直接書き込む場合は `#raw /dev/vg/lv_kdump` 行の先頭にあるハッシュ記号 (「#」) を取り除き、値をダンプ出力先のデバイス名に置き換えます。例を示します。

```
raw /dev/sdb1
```

NFS プロトコルを使ってリモートのマシンにダンプを保存する場合は `#nfs`

`my.server.com:/export/tmp` 行の先頭にあるハッシュ記号 (「#」) を取り除き、値を有効なホスト名とディレクトリパスに置き換えます。

```
nfs penguin.example.com:/export/cores
```

SSH プロトコルを使ってリモートのマシンにダンプを保存する場合は `#ssh user@my.server.com` 行の先頭にあるハッシュ記号 (「#」) を取り除き、値を有効なユーザー名とホスト名に置き換えます。設定に SSH キーも含める場合は `#sshkey /root/.ssh/kdump_id_rsa` 行の先頭にあるハッシュ記号 (「#」) を取り除き、値をダンプ出力先となるサーバー上で有効なキーの場所に変更します。

```
ssh john@penguin.example.com
sshkey /root/.ssh/mykey
```

SSH サーバーの設定方法およびキーベースの認証設定については [『Red Hat Enterprise Linux 7 システム管理者のガイド』](#) を参照してください。

対応しているダンプ出力先と非対応のダンプ出力先のタイプ別一覧は [表B.3 「対応している kdump のダンプ出力先」](#) を参照してください。

2.2.3. コアコレクターを設定する

vmcore ダンプファイルのサイズを小さくするために、**kdump** では外部アプリケーション (コアコレクター) を指定してデータを圧縮、必要に応じて関連性のない情報をすべて除外することができます。現在、完全対応しているコアコレクターは **makedumpfile** のみにあります。

root になりテキストエディターで `/etc/kdump.conf` 設定ファイルを開きます。 `#core_collector makedumpfile -l --message-level 1 -d 31` 行の先頭にあるハッシュ記号 (「#」) を取り除き、以下の説明に従ってコマンドラインのオプションを編集しコアコレクターを使用できるようにします。

ダンプファイルの圧縮を有効にするため `-c` パラメーターを追加します。例を示します。

```
core_collector makedumpfile -c
```

ダンプから特定のページを除外するため `-d value` を追加します。 `value` には [表B.4 「対応しているフィルターレベル」](#) で説明しているように除外対象ページに該当する値の合計を入力します。ゼロで埋められたページと未使用ページを除外する場合は次のようになります。

```
core_collector makedumpfile -d 17 -c
```

使用できるオプションの一覧については **makedumpfile(8)** の man ページをご覧ください。

2.2.4. デフォルト動作を設定する

kdump が [「kdump タイプを設定する」](#) で指定したダンプ出力先でのコアダンプの作成に失敗すると、デフォルトでは **root** ファイルシステムがマウントされコアをローカルに保存しようとします。この動作を変更する場合は **root** になりテキストエディターで `/etc/kdump.conf` 設定ファイルを開きます。 `#default_shell` 行の先頭にあるハッシュ記号 (「#」) を取り除き、[表B.5 「対応しているデフォルトの動作」](#) の説明に従って値を目的の動作に変更します。

例:

```
default reboot
```

2.2.5. サービスを有効にする

起動時に **kdump** デーモンを開始させるには **root** になりシェルプロンプトで以下を入力します。

```
systemctl enable kdump.service
```

multi-user.target のサービスが有効になります。同様に **systemctl stop kdump** を使用するとこのサービスが無効になります。現在のセッションでサービスを開始する場合は **root** になり次のコマンドを使用します。

```
systemctl start kdump.service
```



重要

Red Hat Enterprise Linux 7 では **kdump** のダンプ出力先として指定されているディレクトリーが **kdump systemd** サービスの起動時に存在していなければなりません。起動時にこのディレクトリーが存在しないとサービスの起動は失敗します。この動作は Red Hat Enterprise Linux の旧リリースと異なります。旧リリースではサービスの起動時にこのディレクトリーが存在していないと自動的に作成されていました。

systemd およびサービスの設定方法全般については [『Red Hat Enterprise Linux 7 システム管理者のガイド』](#) を参照してください。

2.3. グラフィカルユーザーインターフェースで kdump を設定する

Kernel Dump Configuration ユーティリティーを起動するにはパネルから **アプリケーション** → **システムツール** → **Kernel crash dumps** の順で選択するか、シェルプロンプトで **system-config-kdump** を入力します。 [図2.1 「基本設定」](#) に示すウィンドウが表示されます。

このユーティリティーを使用すると **kdump** の設定のほか、起動時にサービスを有効または無効にすることもできます。設定が完了したら **適用** をクリックして変更を保存します。認証が済んでいる場合を除きスーパーユーザーのパスワード入力が必要です。また、設定に加えた変更を適用するにはシステムの再起動が必要な旨を示すメッセージが表示されます。

2.3.1. メモリー使用を設定する

基本設定 タブでは **kdump** カーネル用に予約されるメモリーサイズを設定できます。**手動セッティング** のラジオボタンを選択し **新規の kdump メモリー** フィールドの横にある上矢印ボタンまたは下矢印ボタンをクリックして予約するメモリーサイズを増減させます。システムで使用できるメモリーの残量に応じて **使用可能なメモリー** フィールドが変化します。**kdump** のメモリー要件については [「メモリーの要件」](#) を参照してください。

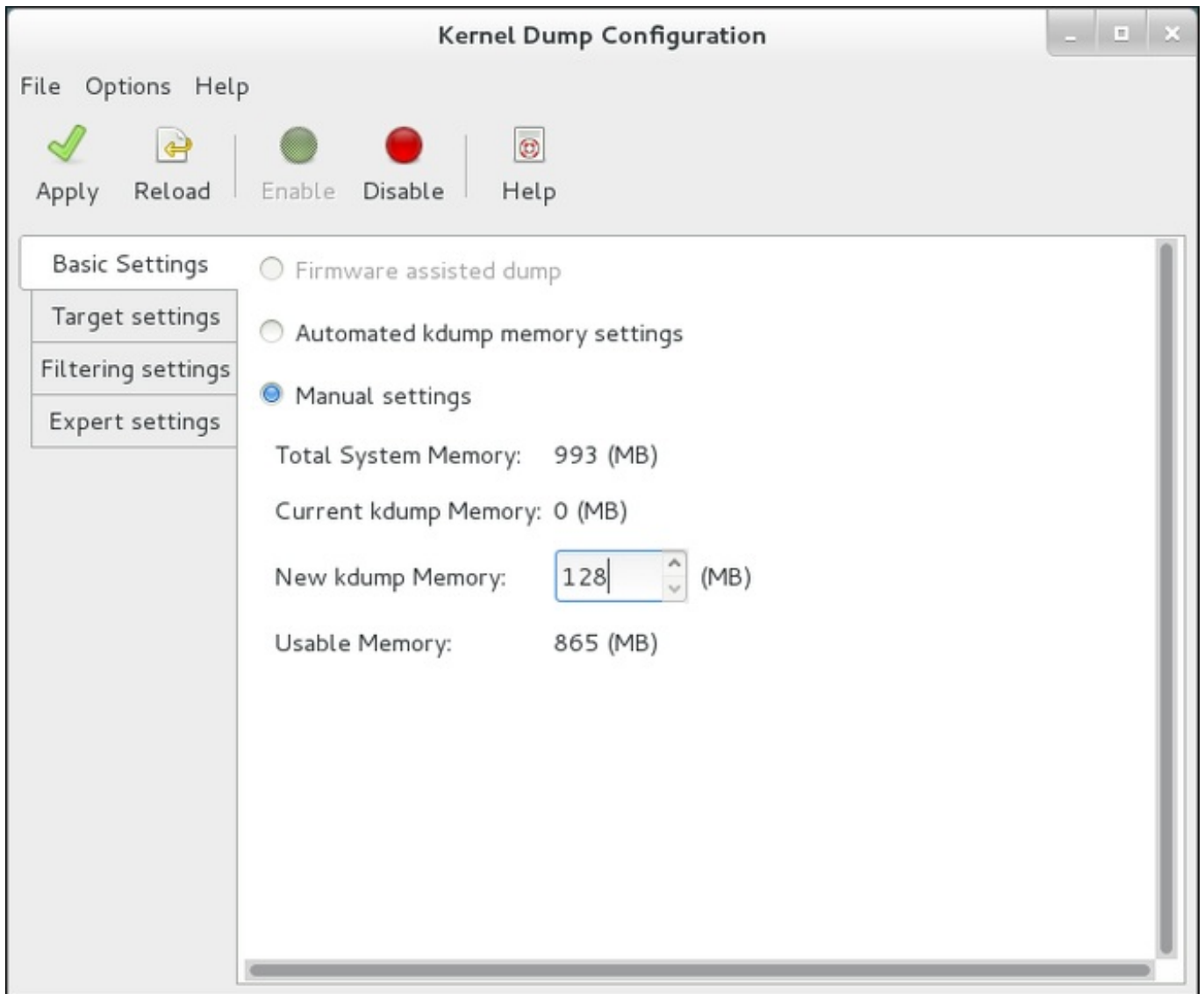


図2.1 基本設定

2.3.2. kdump タイプを設定する

出力先 タブでは **vmcore** ダンプの出力先を指定することができます。ダンプはローカルのファイルシステムにファイルとして保存するか、デバイスに直接書き込む、または **NFS** (Network File System) や **SSH** (Secure Shell) などのプロトコルを使ってネットワーク経由で送信することができます。

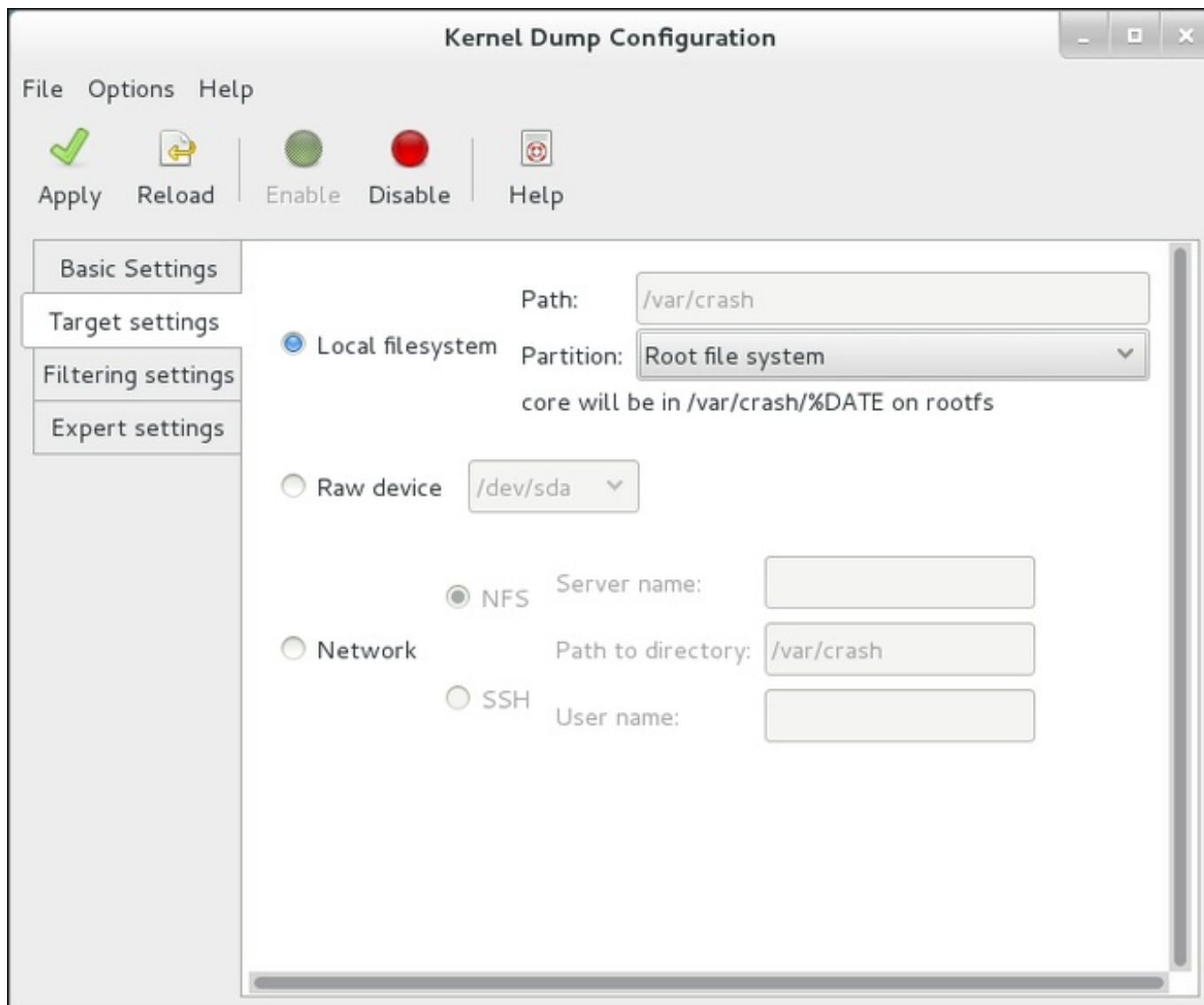


図2.2 出力先

ローカルのファイルシステムにダンプを保存する場合は **ローカルファイルシステム** のラジオボタンを選択します。必要に応じて **パーティション** のドロップダウンリストから別のパーティションを選択、**パス** フィールドを使って出力先ディレクトリーを選択し設定をカスタマイズすることもできます。



重要

Red Hat Enterprise Linux 7 では `kdump` のダンプ出力先として指定されているディレクトリーが **kdump** `systemd` サービスの起動時に存在していなければなりません。起動時にこのディレクトリーが存在しないとサービスの起動は失敗します。この動作は Red Hat Enterprise Linux の旧リリースと異なります。旧リリースではサービスの起動時にこのディレクトリーが存在しないと自動的に作成されていました。

デバイスに直接ダンプを書き込む場合は **Raw デバイス** ラジオボタンを選択し目的の出力先デバイスをドロップダウンリストから選択します。

ネットワーク接続を介してリモートのマシンにダンプを送信する場合は **ネットワーク** ラジオボタンを選択します。**NFS** プロトコルを使用する場合は **NFS** ラジオボタンを選択し **サーバー名** と **ディレクトリーへのパス** フィールドを入力します。**SSH** プロトコルを使用する場合は **SSH** ラジオボタンを選択し **サーバー名**、**ディレクトリーへのパス**、**ユーザー名** のフィールドにリモートサーバーのアドレス、出力先ディレ

クトリー、有効なユーザー名をそれぞれ入力します。

SSH サーバーの設定方法およびキーベースの認証設定については『[Red Hat Enterprise Linux 7 システム管理者のガイド](#)』を参照してください。現在対応している出力先の一覧については表B.3「[対応している kdump のダンプ出力先](#)」を参照してください。

2.3.3. コアコレクターを設定する

フィルタリング タブでは **vmcore** ダンプのフィルターレベルを選択することができます。

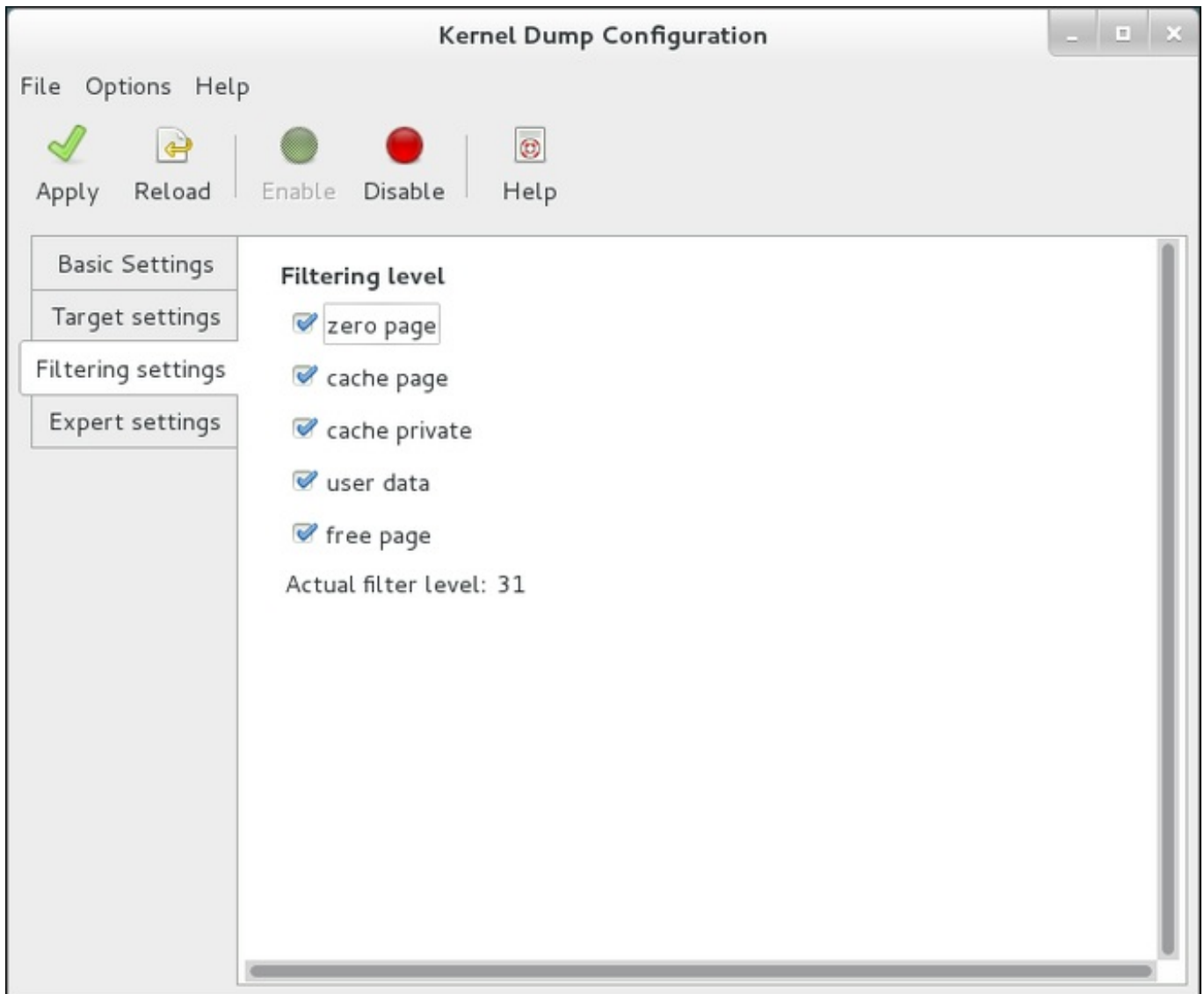


図2.3 フィルタリング

ダンプから **ゼロページ**、**キャッシュページ**、**キャッシュプライベート**、**ユーザーデータ**、または **フリーページ** を除外するには該当ラベルの横にあるチェックボックスを使って選択します。

2.3.4. デフォルト動作を設定する

kdump がコアダンプの作成に失敗した場合に行う動作を選択するには、**ダンプに失敗した場合の動作** のドロップダウンリストから目的のオプションを選択します。**rootfs** にダンプして再起動(コアをローカルに保存してからシステムを再起動するデフォルトの動作)、**reboot** (システムを再起動)、**shell** (対話式シェルプロンプトを表示)、**halt** (システムを停止)、**poweroff** (システムの電源を切断) などのオプションを選択することができます。

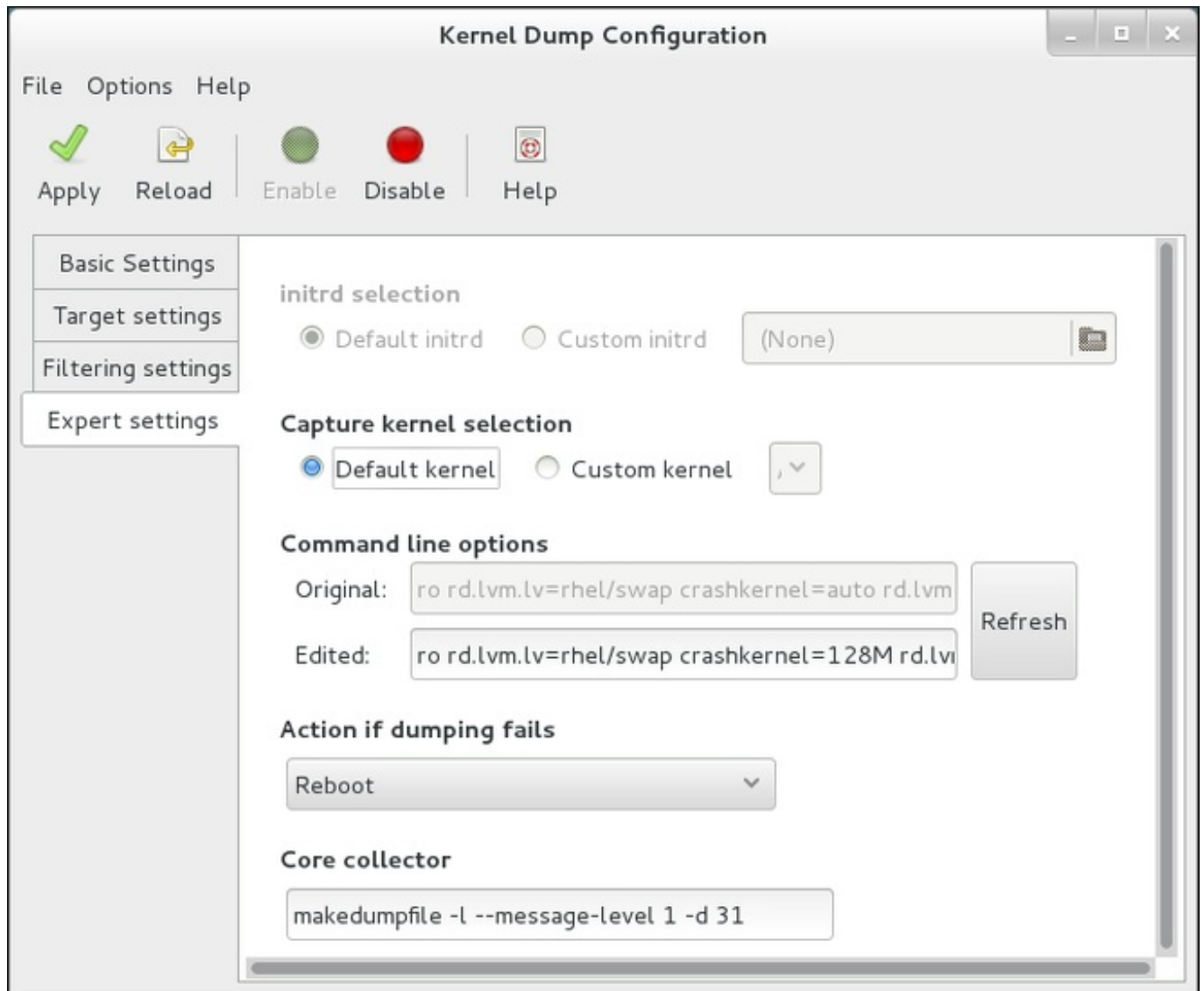


図2.4 フィルタリング

makedumpfile コアコレクターに渡されるオプションをカスタマイズするにはコアコレクター のテキストフィールドを編集します。詳細は [「コアコレクターを設定する」](#) をご覧ください。

2.3.5. サービスを有効にする

起動時に **kdump** サービスを開始する場合はツールバーの**有効化** ボタンをクリックしてから**適用** ボタンをクリックします。**multi-user.target** のサービスが有効になり起動されます。**無効化** ボタンをクリックし**適用** ボタンをクリックするとサービスが直ちに無効になります。



重要

Red Hat Enterprise Linux 7 では **kdump** のダンプ出力先として指定されているディレクトリーが **kdump systemd** サービスの起動時に存在していなければなりません。起動時にこのディレクトリーが存在しないとサービスの起動は失敗します。この動作は Red Hat Enterprise Linux の旧リリースと異なります。旧リリースではサービスの起動時にこのディレクトリーが存在していないと自動的に作成されていました。

systemd のダンプ出力先、サービスの設定方法全般については『[Red Hat Enterprise Linux 7 システム管理者のガイド](#)』を参照してください。

2.4. kdump 設定のテスト



警告

以下のコマンドを実行するとカーネルクラッシュが起こります。次の手順を行う際は十分に注意してください。実稼働のシステムでは絶対に実行しないでください。

設定をテストするため **kdump** を有効にしてシステムを再起動し、サービスが実行されているか確認します。

```
~]# systemctl is-active kdump
active
```

次に、シェルプロンプトで以下のコマンドを入力します。

```
echo 1 > /proc/sys/kernel/sysrq
echo c > /proc/sysrq-trigger
```

このコマンドにより Linux カーネルがクラッシュするため **address-YYYY-MM-DD-HH:MM:SS/vmcore** ファイルが設定で選択した場所にコピーされます (デフォルトでは **/var/crash/**)。

2.5. その他のリソース

2.5.1. インストールされているドキュメント

- ※ **kdump.conf(5)** — **/etc/kdump.conf** 設定ファイルの man ページです。使用できるオプションの詳細な説明が参照できます。
- ※ **zipl.conf(5)** — **/etc/zipl.conf** 設定ファイルの man ページです。
- ※ **zipl(8)** — IBM System z 用 **zipl** ブートローダーユーティリティの man ページです。
- ※ **makedumpfile(8)** — **makedumpfile** コアコレクターの man ページです。
- ※ **kexec(8)** — **kexec** の man ページです。
- ※ **crash(8)** — **crash** ユーティリティの man ページです。
- ※ **/usr/share/doc/kexec-tools-version/kexec-kdump-howto.txt** — **kdump** と **kexec** のインストール方法および使い方に関する簡単な説明が参照できます。

2.5.2. オンラインのドキュメント

<https://access.redhat.com/site/solutions/6038>

kexec と **kdump** の設定に関する Red Hat ナレッジベースの記載です。

<https://access.redhat.com/site/solutions/223773>

対応している **kdump** の出力先に関する Red Hat ナレッジベースの記載です。

<http://people.redhat.com/anderson/>

crash ユーティリティのホームページです。

<https://www.gnu.org/software/grub/>

GRUB2 ブートローダーのホームページとドキュメントが参照できます。

第3章 コアダンプの分析

システムクラッシュの原因を確定する場合、GNU Debugger (GDB) と良く似た対話式のプロンプト **crash** ユーティリティを使用することができます。稼働中の Linux システムだけでなく **netdump**、**diskdump**、**xendump**、**kdump** などで作成したコアダンプなども対話式に分析することができます。

3.1. crash ユーティリティのインストール

crash 分析ツールをインストールするため **root** でシェルプロンプトから次のコマンドを実行します。

```
yum install crash
```

crash に加え *kernel-debuginfo* パッケージもインストールしておく必要があります。このパッケージでダンプ分析に必要なデータが提供されます。このパッケージをインストールするには該当リポジトリを最前位に有効にしておく必要があります。次のコマンドを **root** でコマンドラインから実行します。

```
yum --enablerepo=\*debuginfo
```

リポジトリを有効にしたら次のコマンドを **root** で実行して *kernel-debuginfo* パッケージをインストールします。

```
debuginfo-install kernel
```

Red Hat Enterprise Linux に **Yum** パッケージマネージャーを使って新しいパッケージをインストールする方法については『[Red Hat Enterprise Linux 7 システム管理者のガイド](#)』を参照してください。

3.2. crash ユーティリティの実行

シェルプロンプトで次の形式のコマンドを入力してユーティリティを起動します。

```
crash /var/crash/<timestamp>/vmcore  
/usr/lib/debug/lib/modules/<kernel>/vmlinux
```

<kernel> バージョンは **kdump** でキャプチャーしたバージョンと同じバージョンにしなければならない点に注意してください。現在稼働しているカーネルのバージョンを検索する場合は **uname -r** コマンドを使用します。

例3.1 crash ユーティリティの実行

```
~]# crash /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux \  
/var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore
```

```
crash 5.0.0-23.el6  
Copyright (C) 2002-2010 Red Hat, Inc.  
Copyright (C) 2004, 2005, 2006 IBM Corporation  
Copyright (C) 1999-2006 Hewlett-Packard Co  
Copyright (C) 2005, 2006 Fujitsu Limited  
Copyright (C) 2006, 2007 VA Linux Systems Japan K.K.  
Copyright (C) 2005 NEC Corporation  
Copyright (C) 1999, 2002, 2007 Silicon Graphics, Inc.
```

```
Copyright (C) 1999, 2000, 2001, 2002 Mission Critical Linux, Inc.
This program is free software, covered by the GNU General Public
License,
and you are welcome to change it and/or distribute copies of it under
certain conditions. Enter "help copying" to see the conditions.
This program has absolutely no warranty. Enter "help warranty" for
details.
```

```
GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show
copying"
and "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"...
```

```
      KERNEL: /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux
      DUMPFILE: /var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore [PARTIAL
DUMP]
      CPUS: 4
      DATE: Wed Aug 25 08:44:47 2010
      UPTIME: 00:09:02
LOAD AVERAGE: 0.00, 0.01, 0.00
      TASKS: 140
      NODENAME: hp-dl320g5-02.lab.bos.redhat.com
      RELEASE: 2.6.32-69.el6.i686
      VERSION: #1 SMP Tue Aug 24 10:31:45 EDT 2010
      MACHINE: i686 (2394 Mhz)
      MEMORY: 8 GB
      PANIC: "Oops: 0002 [#1] SMP " (check log for details)
      PID: 5591
      COMMAND: "bash"
      TASK: f196d560 [THREAD_INFO: ef4da000]
      CPU: 2
      STATE: TASK_RUNNING (PANIC)

crash>
```

3.3. メッセージバッファの表示

対話式プロンプトで **log** コマンドを入力しカーネルメッセージバッファを表示させます。

例3.2 カーネルメッセージバッファの表示

```
crash> log
... several lines omitted ...
EIP: 0060:[<c068124f>] EFLAGS: 00010096 CPU: 2
EIP is at sysrq_handle_crash+0xf/0x20
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000
ESI: c0a09ca0 EDI: 00000286 EBP: 00000000 ESP: ef4dbf24
DS: 007b ES: 007b FS: 00d8 GS: 00e0 SS: 0068
```

```

Process bash (pid: 5591, ti=ef4da000 task=f196d560 task.ti=ef4da000)
Stack:
 c068146b c0960891 c0968653 00000003 00000000 00000002 efade5c0
c06814d0
<0> ffffffff c068150f b7776000 f2600c40 c0569ec4 ef4dbf9c 00000002
b7776000
<0> efade5c0 00000002 b7776000 c0569e60 c051de50 ef4dbf9c f196d560
ef4dbfb4
Call Trace:
 [<c068146b>] ? __handle_sysrq+0xfb/0x160
 [<c06814d0>] ? write_sysrq_trigger+0x0/0x50
 [<c068150f>] ? write_sysrq_trigger+0x3f/0x50
 [<c0569ec4>] ? proc_reg_write+0x64/0xa0
 [<c0569e60>] ? proc_reg_write+0x0/0xa0
 [<c051de50>] ? vfs_write+0xa0/0x190
 [<c051e8d1>] ? sys_write+0x41/0x70
 [<c0409adc>] ? syscall_call+0x7/0xb
Code: a0 c0 01 0f b6 41 03 19 d2 f7 d2 83 e2 03 83 e0 cf c1 e2 04 09 d0
88 41 03 f3 c3 90 c7 05 c8 1b 9e c0 01 00 00 00 0f ae f8 89 f6 <c6> 05
00 00 00 00 01 c3 89 f6 8d bc 27 00 00 00 00 8d 50 d0 83
EIP: [<c068124f>] sysrq_handle_crash+0xf/0x20 SS:ESP 0068:ef4dbf24
CR2: 0000000000000000

```

コマンドの詳しい使い方については **help log** を入力します。

3.4. バックトレースの表示

対話式プロンプトで **bt** コマンドを入力しカーネルのスタックトレースを表示させます。1 プロセスのバックトレースを表示させる場合は **bt <pid>** と入力します。

例3.3 カーネルスタックトレースの表示

```

crash> bt
PID: 5591 TASK: f196d560 CPU: 2 COMMAND: "bash"
#0 [ef4dbdcc] crash_kexec at c0494922
#1 [ef4dbe20] oops_end at c080e402
#2 [ef4dbe34] no_context at c043089d
#3 [ef4dbe58] bad_area at c0430b26
#4 [ef4dbe6c] do_page_fault at c080fb9b
#5 [ef4dbee4] error_code (via page_fault) at c080d809
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000 EBP:
00000000
DS: 007b ESI: c0a09ca0 ES: 007b EDI: 00000286 GS:
00e0
CS: 0060 EIP: c068124f ERR: ffffffff EFLAGS: 00010096
#6 [ef4dbf18] sysrq_handle_crash at c068124f
#7 [ef4dbf24] __handle_sysrq at c0681469
#8 [ef4dbf48] write_sysrq_trigger at c068150a
#9 [ef4dbf54] proc_reg_write at c0569ec2
#10 [ef4dbf74] vfs_write at c051de4e
#11 [ef4dbf94] sys_write at c051e8cc
#12 [ef4dbfb0] system_call at c0409ad5

```

```
EAX: ffffffffda  EBX: 00000001  ECX: b7776000  EDX: 00000002
DS: 007b         ESI: 00000002  ES: 007b      EDI: b7776000
SS: 007b         ESP: bfc2088   EBP: bfc20b4  GS: 0033
CS: 0073         EIP: 00edc416  ERR: 00000004  EFLAGS: 00000246
```

コマンドの詳しい使い方については **help bt** を入力します。

3.5. プロセスの状態表示

対話式プロンプトで **ps** コマンドを入力しシステム内のプロセスの状態を表示させます。1 プロセスの状態を表示させる場合は **ps <pid>** と入力します。

例3.4 システム内のプロセスの状態表示

```
crash> ps
  PID   PPID  CPU  TASK           ST  %MEM   VSZ   RSS  COMM
>    0     0    0  c09dc560      RU   0.0     0     0  [swapper]
>    0     0    1  f7072030      RU   0.0     0     0  [swapper]
    0     0    2  f70a3a90      RU   0.0     0     0  [swapper]
>    0     0    3  f70ac560      RU   0.0     0     0  [swapper]
    1     0    1  f705ba90      IN   0.0  2828  1424  init
... several lines omitted ...
 5566     1    1  f2592560      IN   0.0  12876   784  auditd
 5567     1    2  ef427560      IN   0.0  12876   784  auditd
 5587   5132    0  f196d030      IN   0.0  11064  3184  sshd
>  5591   5587    2  f196d560      RU   0.0   5084  1648  bash
```

コマンドの詳しい使い方については **help ps** を入力します。

3.6. 仮想メモリ情報の表示

対話式プロンプトでコマンドを入力し仮想メモリの基本情報を表示させます。1 プロセスの情報を表示させる場合は **vm <pid>** と入力します。

例3.5 現在のコンテキストの仮想メモリ情報の表示

```
crash> vm
PID: 5591  TASK: f196d560  CPU: 2  COMMAND: "bash"
  MM      PGD      RSS      TOTAL_VM
f19b5900  ef9c6000  1648k    5084k
  VMA      START      END      FLAGS  FILE
f1bb0310  242000    260000  8000875  /lib/ld-2.12.so
f26af0b8  260000    261000  8100871  /lib/ld-2.12.so
efbc275c  261000    262000  8100873  /lib/ld-2.12.so
efbc2a18  268000    3ed000  8000075  /lib/libc-2.12.so
efbc23d8  3ed000    3ee000  8000070  /lib/libc-2.12.so
efbc2888  3ee000    3f0000  8100071  /lib/libc-2.12.so
efbc2cd4  3f0000    3f1000  8100073  /lib/libc-2.12.so
efbc243c  3f1000    3f4000  100073
efbc28ec  3f6000    3f9000  8000075  /lib/libdl-2.12.so
```

```

efbc2568    3f9000    3fa000 8100071  /lib/libdl-2.12.so
efbc2f2c    3fa000    3fb000 8100073  /lib/libdl-2.12.so
f26af888    7e6000    7fc000 8000075  /lib/libtinfo.so.5.7
f26aff2c    7fc000    7ff000 8100073  /lib/libtinfo.so.5.7
efbc211c    d83000    d8f000 8000075  /lib/libnss_files-2.12.so
efbc2504    d8f000    d90000 8100071  /lib/libnss_files-2.12.so
efbc2950    d90000    d91000 8100073  /lib/libnss_files-2.12.so
f26afe00    edc000    edd000 4040075
f1bb0a18    8047000   8118000 8001875  /bin/bash
f1bb01e4    8118000   811d000 8101873  /bin/bash
f1bb0c70    811d000   8122000 100073
f26afae0    9fd9000   9ffa000 100073
... several lines omitted ...

```

コマンドの詳しい使い方については **help vm** を入力します。

3.7. オープンファイルの表示

対話式プロンプトで **files** コマンドを入力しオープンファイルに関する情報を表示させます。選択した 1 プロセスで開いているファイルを表示させる場合は **files <pid>** と入力します。

例3.6 現在のコンテキストのオープンファイルについての情報の表示

```

crash> files
PID: 5591   TASK: f196d560   CPU: 2   COMMAND: "bash"
ROOT: /    CWD: /root
FD   FILE      DENTRY      INODE      TYPE      PATH
 0   f734f640  eedc2c6c    eecd6048   CHR      /pts/0
 1   efade5c0  eee14090    f00431d4   REG      /proc/sysrq-trigger
 2   f734f640  eedc2c6c    eecd6048   CHR      /pts/0
10   f734f640  eedc2c6c    eecd6048   CHR      /pts/0
255  f734f640  eedc2c6c    eecd6048   CHR      /pts/0

```

コマンドの詳しい使い方については **help files** を入力します。

3.8. ユーティリティの終了

exit または **q** を入力して対話式プロンプトを閉じ **crash** を終了します。

例3.7 crash ユーティリティの終了

```

crash> exit
~]#

```

付録A よくある質問

問：

Red Hat サポートサービスに大きなサイズのダンプをアップロードする場合はどうしたらいいですか？

答： 分析のため Red Hat グローバルサポートサービスにカーネルクラッシュのダンプファイルを送信していただくかなければならない場合があります。ただし、ダンプファイルのサイズはフィルターで特定の情報に絞り込んだ場合でも非常に大きくなる可能性があります。新しいサポートケースを開いた場合、250 MB を超えるファイルは Red Hat カスタマーポータルからは直接アップロードしていただくことができないため、Red Hat では大きなサイズのファイルのアップロード用に FTP サーバーを用意しています。

FTP サーバーのアドレスは **dropbox.redhat.com** になります。ファイルは **/incoming/** ディレクトリーにアップロードしてください。ご使用の FTP クライアントを **passive** モードに設定しておく必要があります。ご使用のファイアウォールでこのモードを許可していない場合は **origin-dropbox.redhat.com** サーバーを **active** モードで使用していただくことができます。

アップロードするファイルは必ず **gzip** などで圧縮し、ファイル名にはわかりやすい名前を付けてください。ファイル名にサポートケース番号を使用されることをお勧めします。必要なファイルをすべてアップロードしたらサポートケース担当のエンジニアへ正確なファイル名とその SHA1 または MD5 チェックサムをお知らせください。

詳細な説明については <https://access.redhat.com/site/solutions/2112> をご覧ください。

付録B kdump で対応している設定とダンプ出力先

B.1. kdump メモリ要件

kdump でカーネルクラッシュダンプをキャプチャーして分析用に保存しておくためにはシステムメモリーの一部をキャプチャーカーネル用に永続的に予約しておかなければなりません。システムのアーキテクチャーに準じて kdump 用として必要になる最小メモリーと使用可能な物理メモリー合計を以下に示します。

コマンドラインでメモリー設定を変更する方法については [「メモリー使用を設定する」](#) を参照してください。グラフィカルユーザーインターフェースで予約メモリーの設定を変更する方法については [「メモリー使用を設定する」](#) を参照してください。

表B.1 kdump 用に必要な最小予約メモリー

アーキテクチャー	使用可能なメモリー	最小予約メモリー
AMD64 と Intel 64 (x86_64)	2 GB 以上	4 KB の RAM ごと 160 MB + 2 ビット、メモリーが 1 TB を超えるシステムの場合は 224 MB が最小 (160 + 64 MB)
IBM POWER (ppc64)	2 GB から 4 GB	256 MB の RAM
	4 GB から 32 GB	512 MB の RAM
	32 GB から 64 GB	1 GB の RAM
	64 GB から 128 GB	2 GB の RAM
	128 GB 以上	4 GB の RAM
IBM System z (s390x)	2 GB 以上	4 KB の RAM ごと 160 MB + 2 ビット、メモリーが 1 TB を超えるシステムの場合は 224 MB が最小 (160 + 64 MB)

B.2. メモリー自動予約の最小しきい値

一部のシステムではブートローダーの設定ファイルで `crashkernel=auto` パラメーターを使用したり、グラフィカル設定ユーティリティーで自動割り当ての設定を有効にすると kdump 用のメモリー割り当てを自動的に行わせることができます。ただし、この自動予約が正常に機能するためにはシステムで使用できるメモリーの合計が特定のサイズ必要になります。このサイズはシステムのアーキテクチャーによって異なります。

自動メモリー割り当てのしきい値を以下に示します。システムのメモリーが以下に示すしきい値を下回る場合は手作業でメモリー予約を行う必要があります。

コマンドラインで設定を変更する方法については [「メモリー使用を設定する」](#) を参照してください。グラフィカルユーザーインターフェースで予約メモリーのサイズを変更する方法については [「メモリー使用を設定する」](#) を参照してください。

表B.2 自動メモリー予約に必要な最小メモリーサイズ

アーキテクチャー	必要なメモリー
AMD64 と Intel 64 (x86_64)	2 GB
IBM POWER (ppc64)	2 GB
IBM System z (s390x)	4 GB

B.3. 対応している kdump のダンプ出力先

カーネルクラッシュをキャプチャーする際、コアダンプを直接デバイスに書き込んでローカルファイルシステムにファイルとして保存するかネットワーク経由で送信することができます。現在対応しているダンプ出力先および kdump による非対応が明確なダンプ出力先の全一覧を以下に示します。

コマンドラインでダンプ出力先を設定する方法については「[kdump タイプを設定する](#)」を参照してください。グラフィカルユーザーインターフェースでダンプ出力先を設定する方法については「[kdump タイプを設定する](#)」を参照してください。

表B.3 対応している kdump のダンプ出力先

タイプ	対応しているダンプ出力先	非対応のダンプ出力先
raw デバイス	ローカルで添付されたすべての生デバイスとパーティション	—
ローカルファイルシステム	ディスクドライブ、ハードウェア RAID の論理ドライブ、LVM デバイスおよび mdraid アレイに直接接続している ext2、ext3、ext4、btrfs、xfs の各ファイルシステム	この表で明示的にサポート対象として記載されていないローカルのファイルシステム、auto タイプなど (自動ファイルシステム検出)
リモートディレクトリー	<p>IPv4 経由で NFS や SSH プロトコルを使ってアクセスするリモートディレクトリー</p> <p>ソフトウェアのインシエーター経由で iSCSI プロトコルを使ってアクセスするリモートディレクトリー (iBFT (iSCSI Boot Firmware Table) を使用しない)</p> <p>マルチパススペースのストレージ</p> <p>—</p>	<p>NFS プロトコルを使ってアクセスする rootfs ファイルシステムのリモートディレクトリー</p> <p>iBFT を使用し iSCSI プロトコルでアクセスするリモートディレクトリー</p> <p>ハードウェアのインシエーター経由で iSCSI プロトコルを使ってアクセスするリモートディレクトリー</p> <p>IPv6 経由でアクセスするリモートディレクトリー</p> <p>SMB/CIFS プロトコルを使ってアクセスするリモートディレクトリー</p> <p>FCoE (Fibre Channel over Ethernet) プロトコルを使ってアクセスするリモートディレクトリー</p> <p>ワイヤレスネットワークインターフェースを使ってアクセスするリモートディレクトリー</p>

B.4. 対応している kdump のフィルターレベル

ダンプファイルのサイズを縮小させるため kdump では makedumpfile コアコレクターを使ってデータを圧縮し必要に応じて関連性のない情報を除外します。現在、makedumpfile ユーティリティーで対応しているフィルターのレベル全一覧を以下に示します。

コマンドラインでコアコレクターを設定する方法については「[コアコレクターを設定する](#)」を参照してください。グラフィカルインターフェースでコアコレクターを設定する方法については「[コアコレクターを設定する](#)」を参照してください。

表B.4 対応しているフィルターレベル

オプション	説明
1	ゼロページ
2	キャッシュページ
4	キャッシュプライベート
8	ユーザーページ
16	フリーページ

B.5. 対応しているデフォルトの動作

kdump がコアダンプの作成に失敗するとデフォルトでは root ファイルシステムをマウントしてコアのローカルへの保存を試行します。第 1 ダンプ出力先へのコアダンプの保存に失敗した場合、kdump に別の動作を行うよう設定することができます。kdump で現在対応しているデフォルト動作を以下に示します。

コマンドラインでデフォルト動作を設定する方法については「[デフォルト動作を設定する](#)」を参照してください。グラフィカルユーザーインターフェースでデフォルト動作を設定する方法については「[デフォルト動作を設定する](#)」を参照してください。

表B.5 対応しているデフォルトの動作

オプション	説明
dump_to_rootfs	コアダンプの root ファイルシステムへの保存を試行します。ネットワーク上のダンプ出力先と併用する場合に特に便利なオプションです。ネットワーク上のダンプ出力先にアクセスできない場合、ローカルにコアダンプを保存するよう kdump の設定を行います。ダンプ後システムは再起動されます。
reboot	システムを再起動します。コアダンプは失われます。
halt	システムを停止します。コアダンプは失われます。
poweroff	システムの電源を切ります。コアダンプは失われます。
shell	initramfs 内からシェルセッションを実行しユーザーが手作業でコアダンプを記録できるようにします。

付録C 改訂履歴

改訂 1.2-0.2 翻訳および査読完了	Mon Aug 10 2015	Noriko Mizumoto
改訂 1.2-0.1 翻訳ファイルを XML ソースバージョン 1.2-0 と同期	Thu Aug 6 2015	Noriko Mizumoto
改訂 1.2-0 メモリー設定に関する誤った記載および古いスクリーンショットなどを修正するため更新	Fri 06 Mar 2015	Petr Bokoč
改訂 1.1-3 Red Hat Enterprise Linux 7.1 GA リリースのカーネルクラッシュダンプガイド	Wed 18 Feb 2015	Petr Bokoč
改訂 1.1-0 Red Hat Enterprise Linux 7.1 Beta リリースのカーネルクラッシュダンプガイド	Fri 05 Dec 2014	Petr Bokoč
改訂 1.0-0 Red Hat Enterprise Linux 7.0 GA リリースのカーネルクラッシュダンプガイド	Mon 02 Jun 2014	Jaromír Hradílek
改訂 0.0-8 初版作成	Thu Jan 17 2013	Jaromír Hradílek