

# Linux

Viエディタ

11

時間目

# viエディタとは

## 解説

Linuxでは、設定ファイル等をはじめ、多くの場面で文字情報が記録されたテキストファイルを編集する場面に遭遇します。テキストファイルを編集するためのソフトをテキストエディタと言いますが、Linuxでは標準のテキストエディタとして「viエディタ」が搭載されています。ここではviエディタの操作方法を学んでいきます。

## 【参考】 代表的なテキストエディタ

### Windows

- ・ メモ帳
- ・ ワードパッド

### MacOS X

- ・ テキストエディット
- ・ TextMate

### Linux

- ・ **viエディタ**
- ・ Emacs

# viエディタの特徴

## 解説

viエディタは、Windowsのメモ帳などと違って、CUI上（コマンド）で操作するため、以下のように viエディタならではの特徴があります。

## 【特徴】

- 1 Linuxに標準で搭載されているテキストエディタである
- 2 「コマンドモード」と「編集モード」の二つのモードを使い分けて操作する
- 3 操作になればGUIのテキストエディタ(メモ帳など)より素早く操作できる

# viエディタのモード

## 解説

CUIで動作するviエディタは、Windowsのメモ帳のようにメニューバーを表示することが出来ません。そのかわりにコマンドモードと挿入モードを切り替えて操作します。各モードの特徴は以下の通りです。

## コマンドモード

- viエディタを起動した直後は、コマンドモードになっている
- コマンドモードは文字入力できない
- カーソルの移動、文字入力の切り替え、ファイルの保存、コピー貼り付けなどをすべてコマンドで操作する

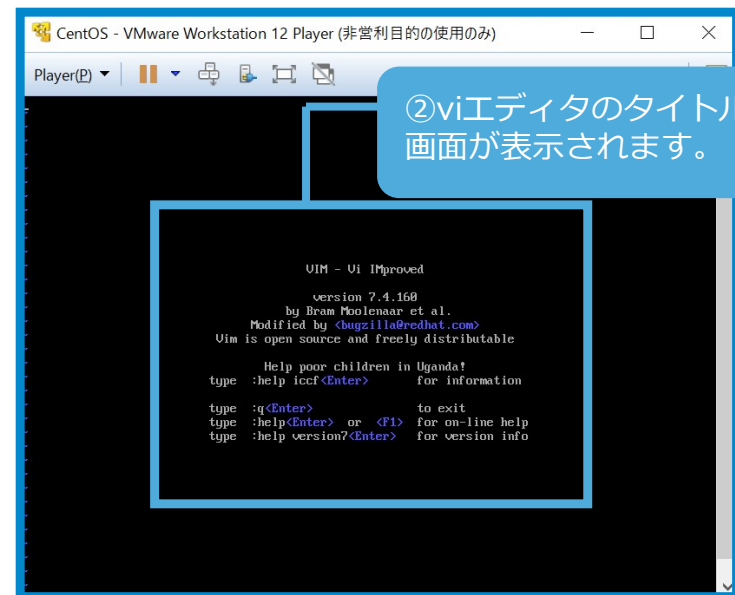
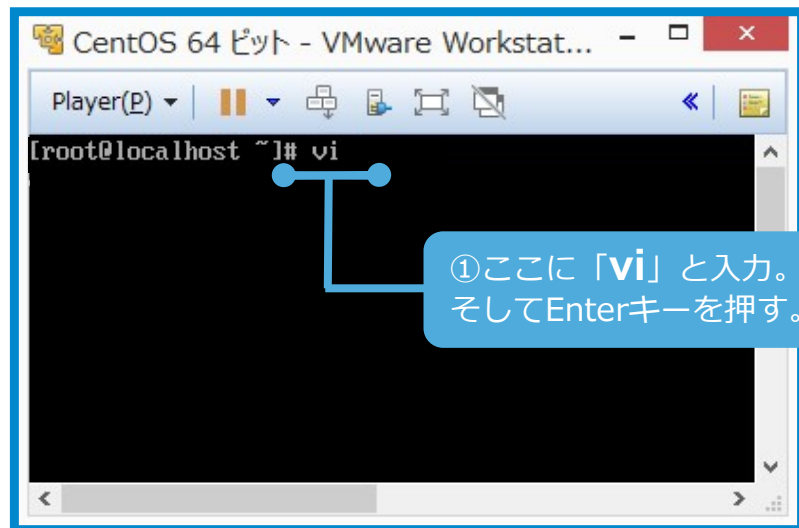
## 編集モード

- 文字を入力するときは、編集モードに切り替える（例：「i」キーを押すと入力モードに切り替えれる）
- 編集モードに切り替えるキーは何種類があり、カーソルの切り替え位置でそれぞれ特徴が違う
- コマンドモードに戻るときは「Esc」キーを押す

## viエディタのタイトル画面①

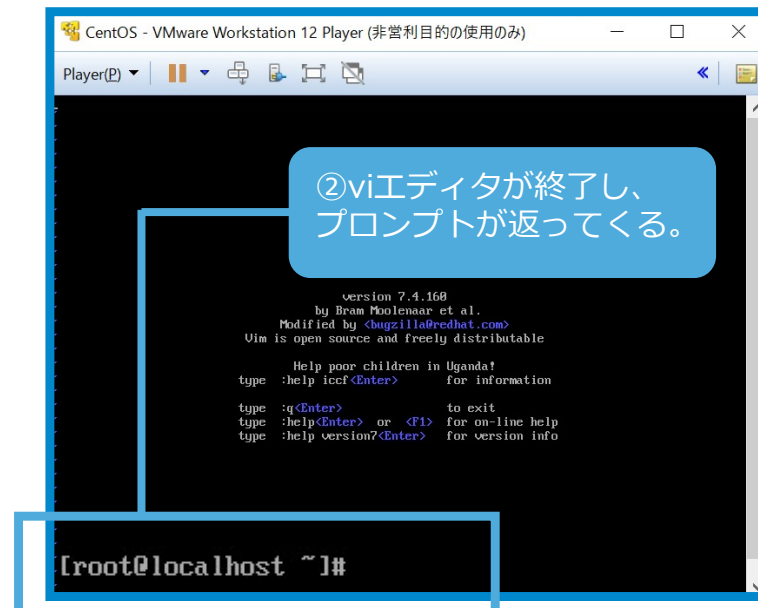
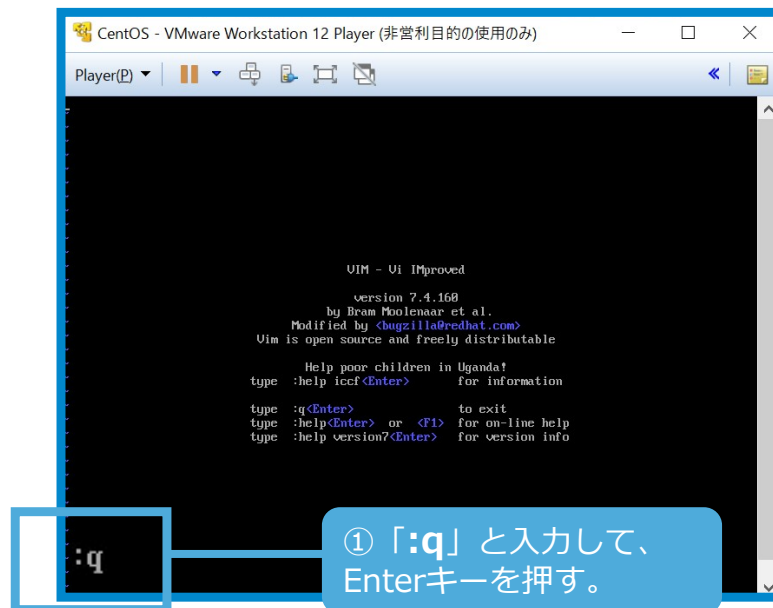
viエディタを起動してみましょう

8 vi ファイル名 テキストファイルを編集する



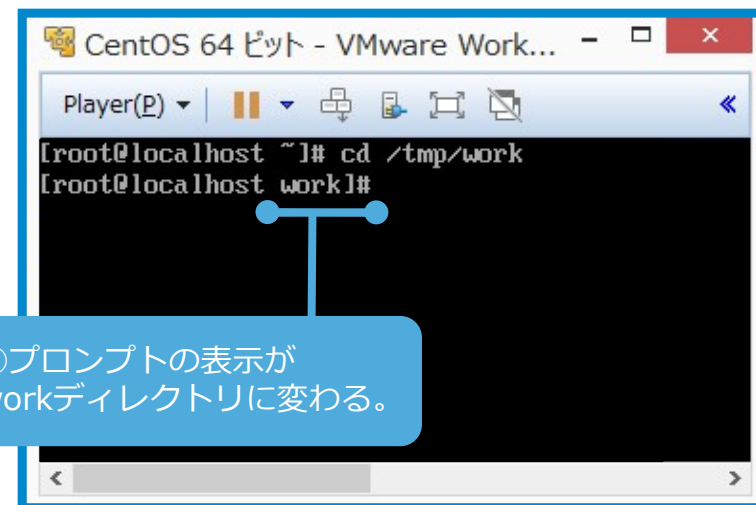
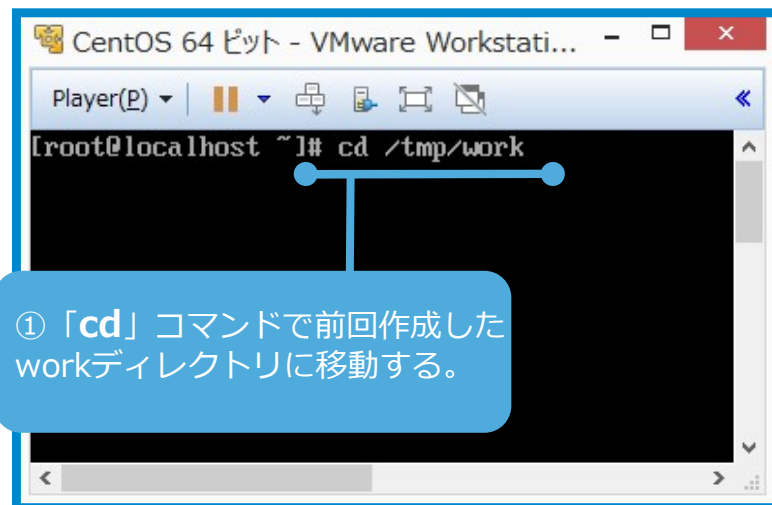
## viエディタのタイトル画面②

viエディタを終了してみましょう



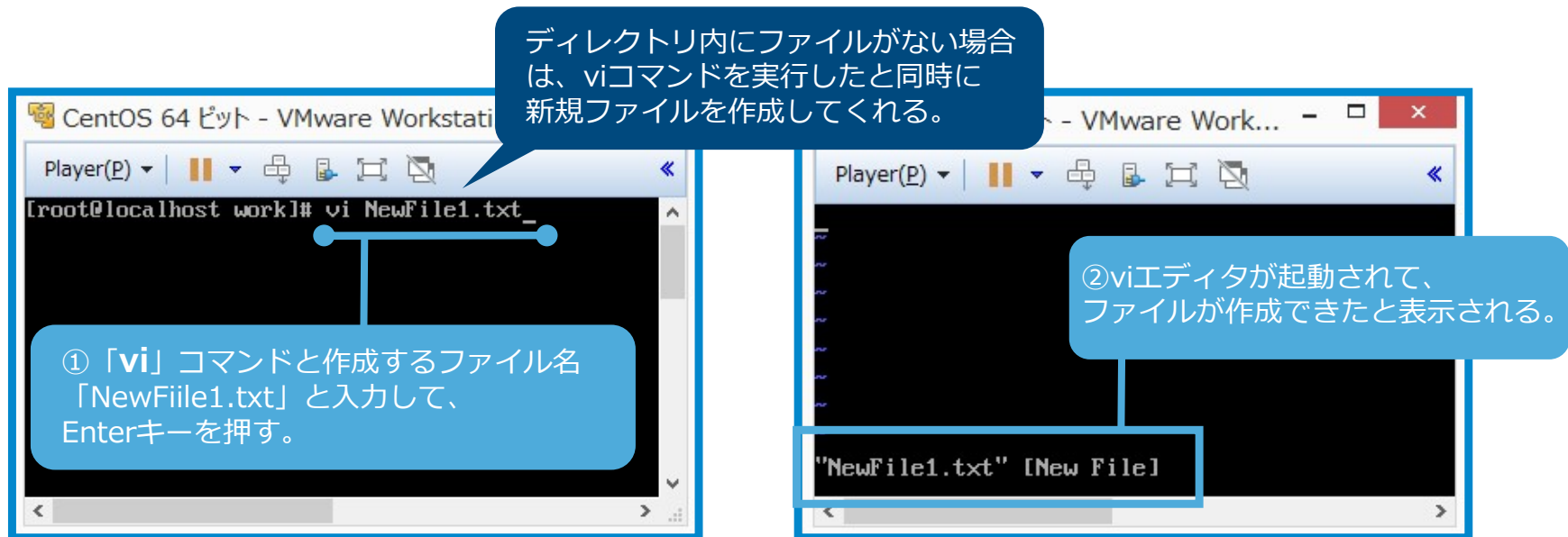
# viエディタでファイルの新規作成①

まずファイルを作るディレクトリに移動



## viエディタでファイルの新規作成②

### 新規でファイルを作成する



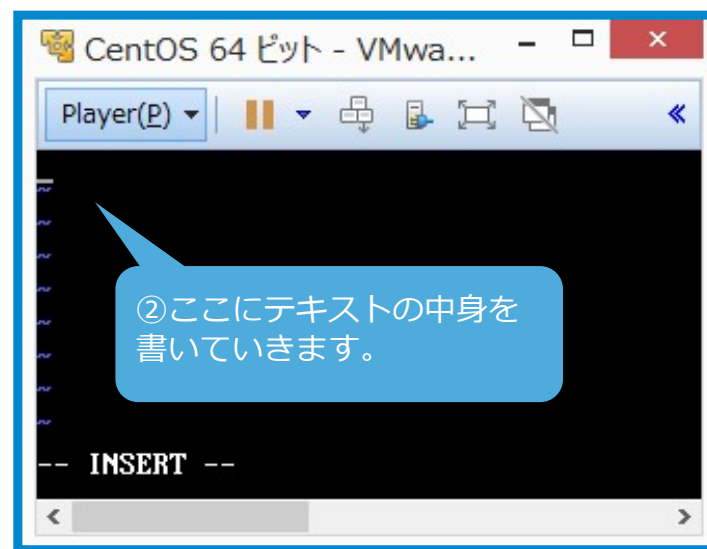
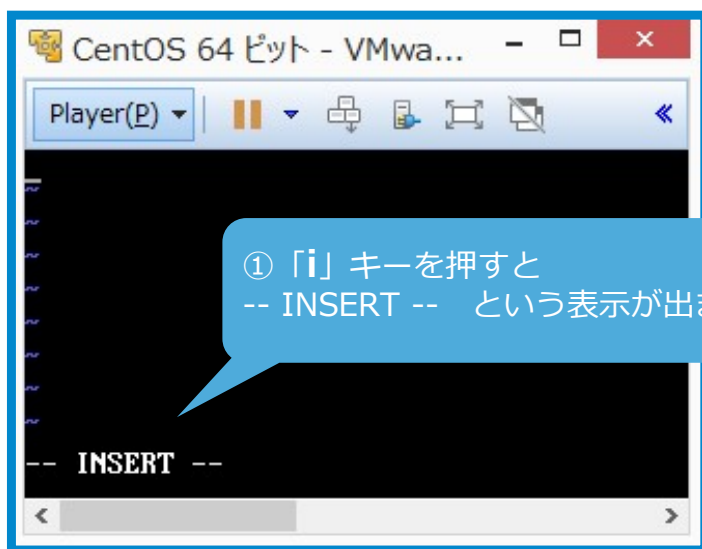


# 編集モードに切り替える

解説

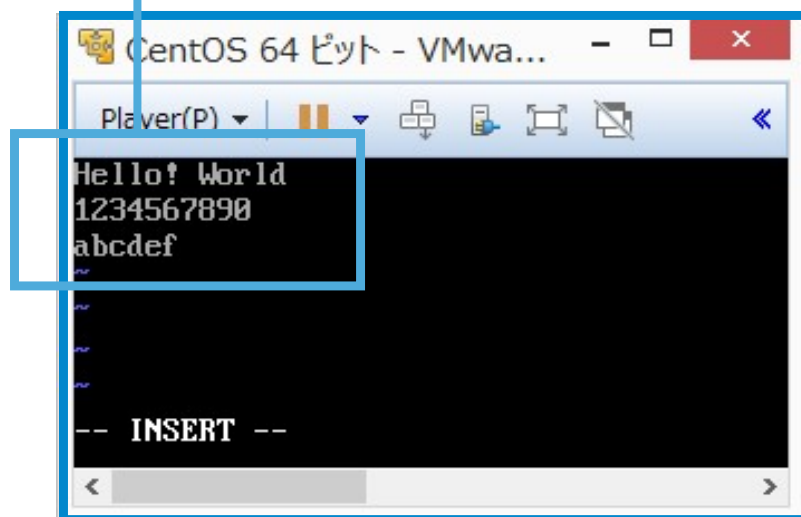
viエディタを起動した直後はコマンドモードになっているため、このままでは文字を入力することが出来ません。  
「i」キーを押して、文字を入力できる状態にしましょう。

## 編集モードに切り替える



# ファイルの中身を記述する

**Hello World**  
**1234567890**  
**abcdef**  
書いてみます。



## <ワンポイント>

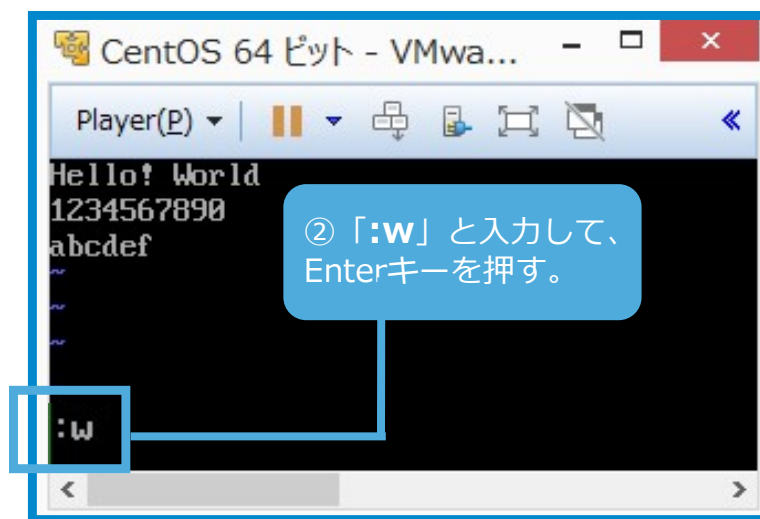
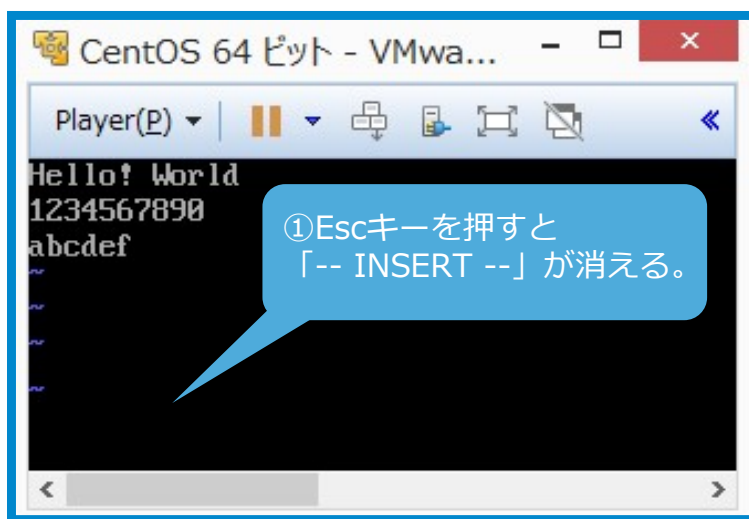
- 1 カーソルの移動は矢印キーで行う。
- 2 改行はEnterキー行う。
- 3 文字の削除は「BackSpace」キー、または「Del」キーで行う。

# ファイルを保存する

## 解説

文章の編集が終わった後、コマンドモードに戻るには「Esc」キーを押します。  
※コピー&貼り付け、ファイルの保存、viエディタの終了など、文字入力以外の操作はコマンドモードに切り替える必要があります。

## 「:w」でファイルを保存する

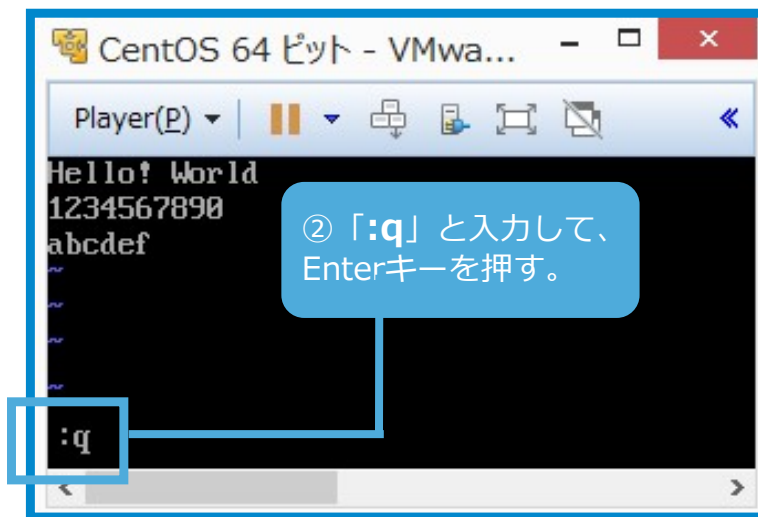
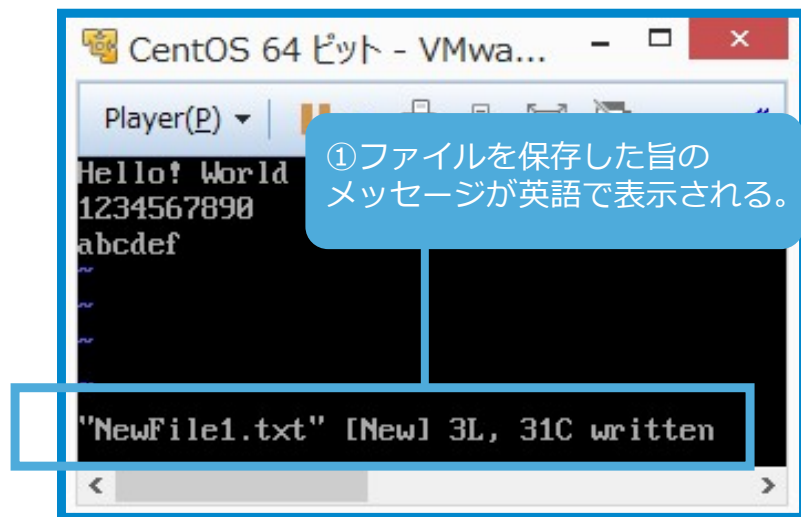


# viエディタを終了する

## 解説

viエディタを終了するには「:q」と入力します。  
ファイルの保存とviエディタの終了を同時に実行するには「:wq」と入力してEnterキーを押します。

## 「:q」でviエディタを終了する

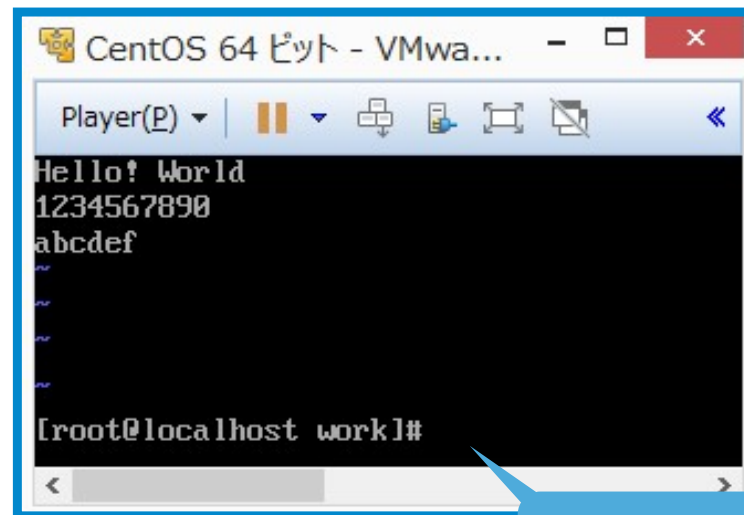


## 補足

編集した後、保存しないで強制的に終了する時は「:q!」と入力してEnterキーを押します。

## viエディタを終了する

Viエディタが終了してプロンプトが表示される。



```
CentOS 64 ビット - VMwa...  
Player(P) | [Icons]  
Hello! World  
1234567890  
abcdef  
[root@localhost work]#
```

また通常の操作ができる  
ようになります。

# モードの切り替えコマンド

## 解説

編集モードへの切り替えは「i」キーだけではなく、いくつか方法があります。  
覚えておくと効率よく編集モードへ切り替えることが可能です。

入力操作	意味	備考
「i」キー	カーソルの位置から入力する。	左記の操作はコマンドモードのみで操作可能。
「I」キー ※大文字のI	カーソルがある行の行頭から入力する。	
「o」キー	カーソルのある行の下に空白行を挿入し、その空白行の行頭から入力する。	
「O」キー ※大文字のO	カーソルのある行の上に空白行を挿入し、その空白行の行頭から入力する。	
「a」キー	カーソルの右側の位置から入力する。	
「A」キー ※大文字のA	カーソルを行末に移動して、その右の位置から入力する。	

# viエディタ カーソル移動コマンド一覧

解説

コマンドモードでカーソルを移動させる方法は以下の通りです。覚えておくと便利です。

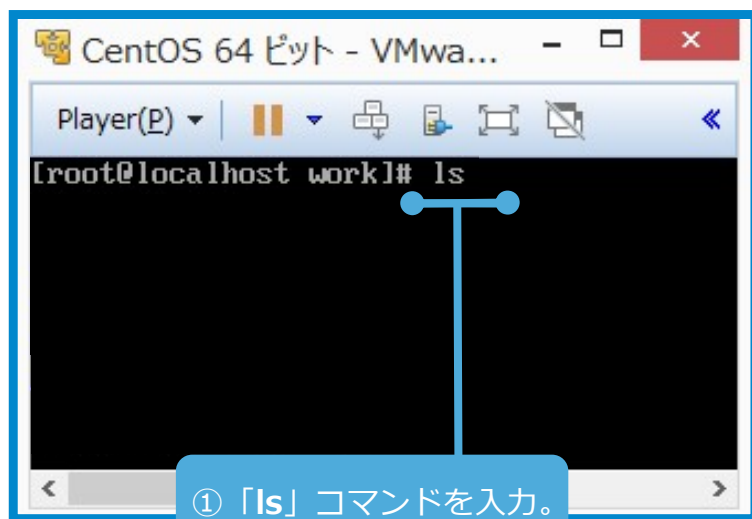
入力操作	意味	備考
「 <b>h</b> 」キー、「←」キー	左に1文字分移動する。	・「h」「j」「k」「l」キーは コマンドモードのみで操作可能。 ・「←」キーはコマンドモード、挿入モードの 両方で操作可能。
「 <b>j</b> 」キー、「↓」キー	下に1文字分移動する。	
「 <b>k</b> 」キー、「↑」キー	上に1文字分移動する。	
「 <b>l</b> 」キー、「→」キー	右に1文字分移動する。	
「 <b>G</b> 」キー ※大文字のG	ファイルの末尾（一番最後）に移動する。	
「 <b>gg</b> 」 ※「 <b>g</b> 」キーを2回	ファイルの先頭に移動する。	左記の操作はコマンドモードのみで操作可能。
「 <b>Ctrl</b> 」キー+「 <b>f</b> 」キー	下に1画面分だけ移動する。	
「 <b>Ctrl</b> 」キー+「 <b>b</b> 」キー	上に1画面分だけ移動する。	

# 作成したファイルを確認する

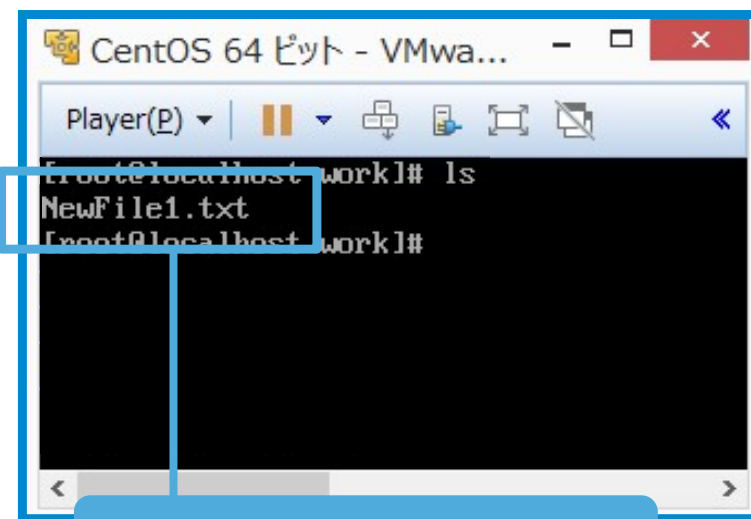
解説

先ほど作ったファイルが保存されていることを確認してみましょう。確認は「ls」コマンドを使用します。  
※「ls」コマンドは8時間目を確認してください。

## 3 ls ファイルやディレクトリのリストを表示する



① 「ls」 コマンドを入力。



② 作成したファイルが表示されます。

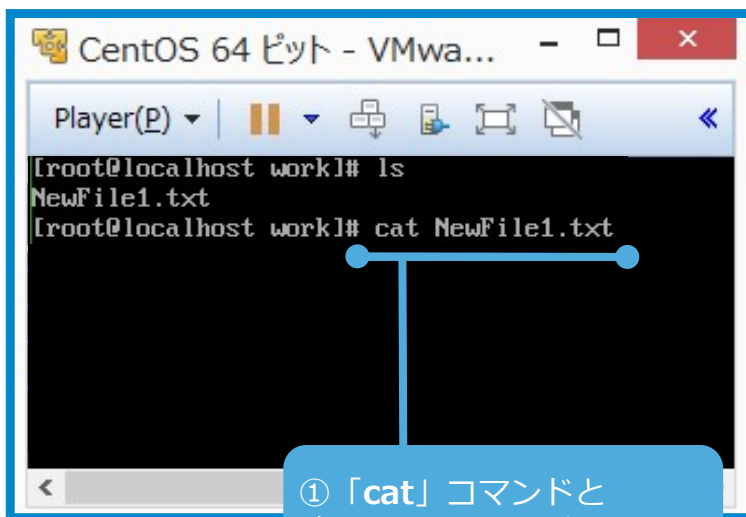


# 作成したファイルの中身を確認する

解説

先ほど作ったファイルの中身が記述され、保存されていることを確認してみましょう。  
ファイルの中身の確認は「cat」コマンドを使用します。

## 9 cat ファイルの内容を表示する

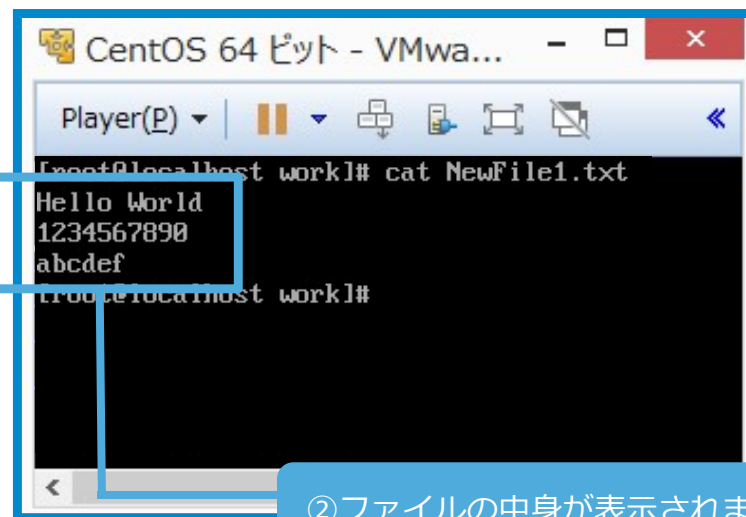


A terminal window titled "CentOS 64 ビット - VMwa..." showing the following commands and output:

```
[root@localhost work]# ls
NewFile1.txt
[root@localhost work]# cat NewFile1.txt
```

A blue line connects the command `cat NewFile1.txt` to a callout box below.

①「cat」コマンドと  
確認するファイル名を入力。



A terminal window titled "CentOS 64 ビット - VMwa..." showing the output of the `cat` command:

```
[root@localhost work]# cat NewFile1.txt
Hello World
1234567890
abcdef
[root@localhost work]#
```

A blue box highlights the output lines, which are connected by a line to a callout box below.

②ファイルの中身が表示されます。

# コピー、貼り付け、切り取りコマンド

## 解説

文章の編集でよく使うコピー、貼り付け、切り取りはコマンドモードで行います。

入力操作	意味	備考
dd ※「d」キーを2回押す	カーソルのある行を切り取る。 ※「3dd」と入力すれば、カーソルのある行から下三行を切り取ることができる。	左記の操作はコマンドモードのみで操作可能。 ※編集モードでは左記のコマンドは使えない。
yy ※「y」キーを2回押す	カーソルのある行をコピーする。 ※「3yy」と入力すれば、カーソルのある行から下三行をコピーすることができる。	
p ※「p」キーを1回押す	カーソルのある行の次の行に貼り付ける。	
x ※「x」キーを1回押す	カーソルの位置にある1文字を切り取る。	

# コピー & 貼り付け①

解説

コピーと貼り付けの練習をします。以下の手順で進めていきます。

- 1 viエディタを起動してファイル「NewFile1.txt」を開きます。  
【コマンド】 **vi NewFile1.txt**



- 2 1行目にカーソルを合わせて「Hello! World」をコピーする。  
【コマンド】 **yy**

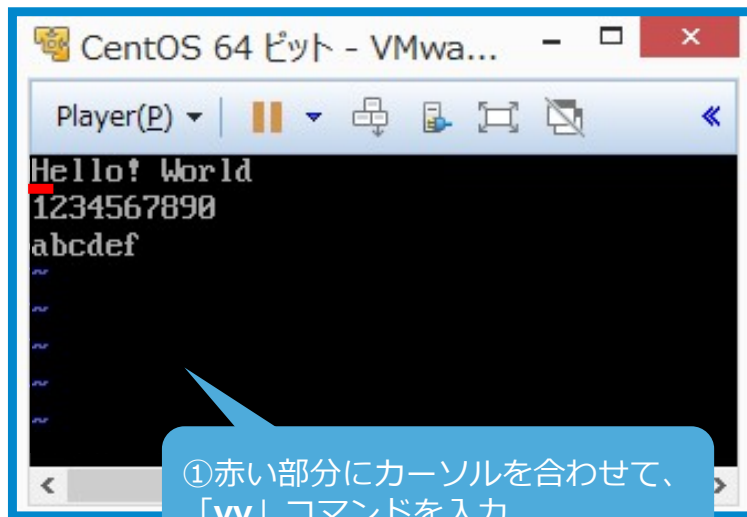


- 3 最後の行にカーソルを合わせて「Hello! World」を貼り付ける。  
【コマンド】 **p**

## コピー & 貼り付け②

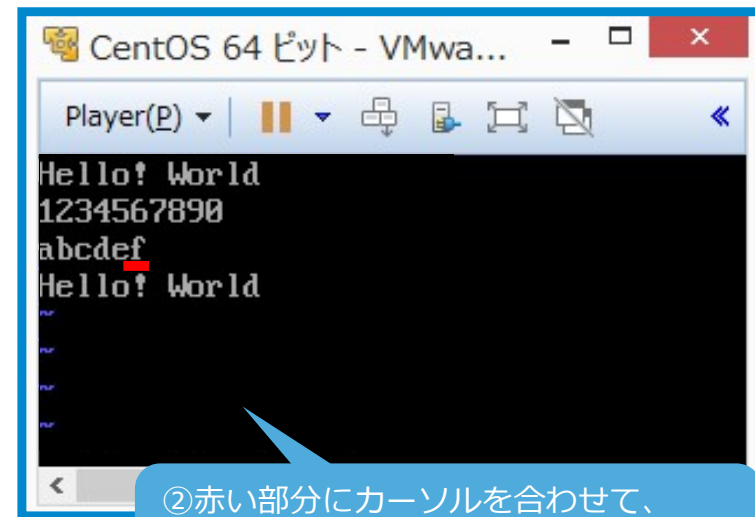
解説

コピーと貼り付けの練習をします。以下のコマンドを実行して下さい。



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
Hello! World
1234567890
abcdef
```

①赤い部分にカーソルを合わせて、「yy」コマンドを入力。



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
Hello! World
1234567890
abcdef
Hello! World
```

②赤い部分にカーソルを合わせて、「p」コマンドを入力すると、「Hello! World」が貼り付けられます。

# 切り取り & 貼り付け①

解説

切り取りと貼り付けを練習します。以下のコマンドを実行して下さい。

- 1 2行目にカーソルを合わせて「1234567890」を切り取る。  
【コマンド】 **dd**



- 2 最後の行にカーソルを合わせて「1234567890」を貼り付ける。  
【コマンド】 **p**

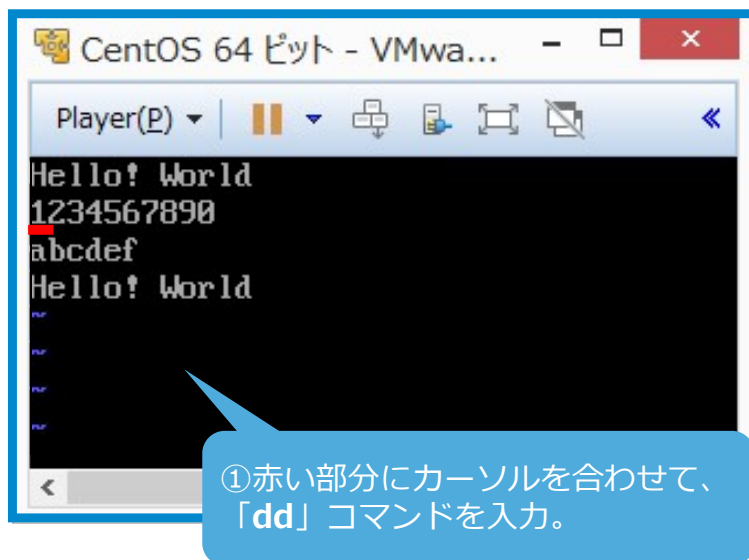


- 3 ファイル「NewFile1.txt」を保存して、viエディタを終了する。  
【コマンド】 **:wq**

## 切り取り & 貼り付け②

解説

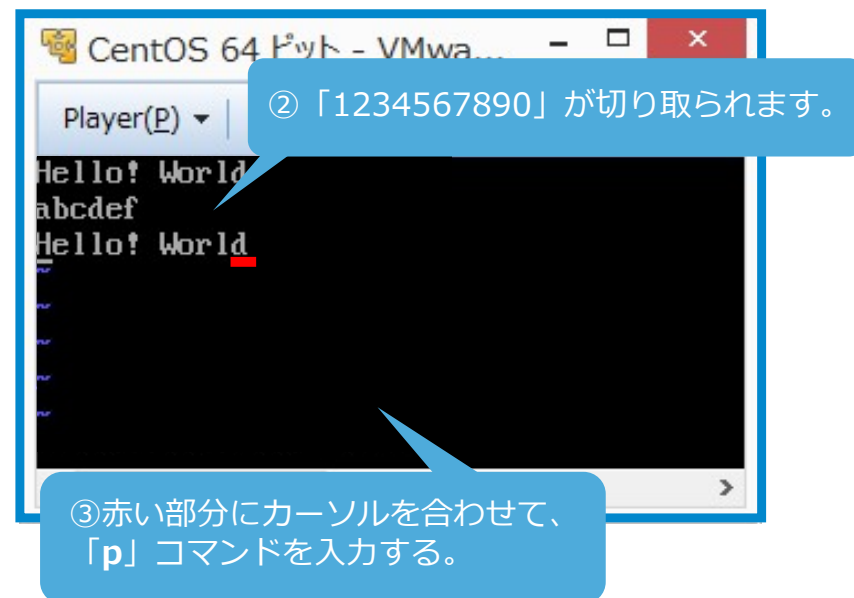
切り取りと貼り付けの練習をします。以下のコマンドを実行して下さい。



The screenshot shows a terminal window titled "CentOS 64 ビット - VMwa...". The terminal content is as follows:

```
Hello! World  
1234567890  
abcdef  
Hello! World
```

A red cursor is positioned at the start of the second line, "1234567890". A blue callout bubble points to this cursor with the text: ①赤い部分にカーソルを合わせて、「dd」コマンドを入力。



The screenshot shows the same terminal window as the previous one, but the content has changed:

```
Hello! World  
abcdef  
Hello! World
```

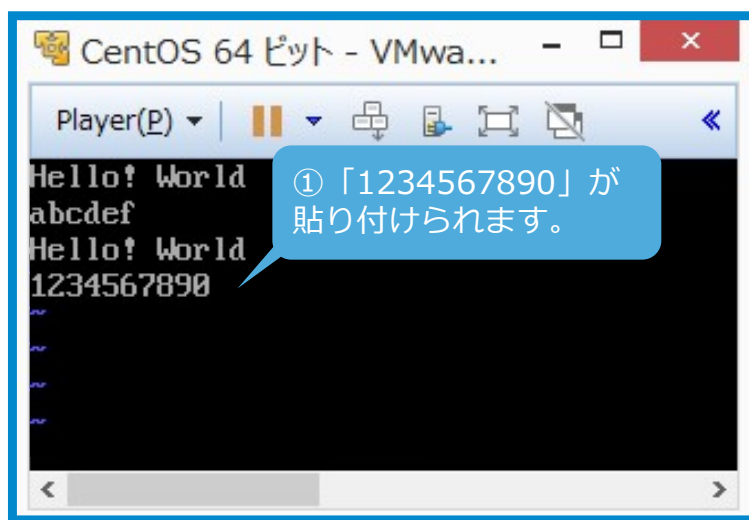
The second line, "1234567890", has been removed. A red cursor is now at the start of the third line, "Hello! World". A blue callout bubble points to this cursor with the text: ③赤い部分にカーソルを合わせて、「p」コマンドを入力する。

Another blue callout bubble points to the removed line area with the text: ②「1234567890」が切り取られます。

## 切り取り & 貼り付け③

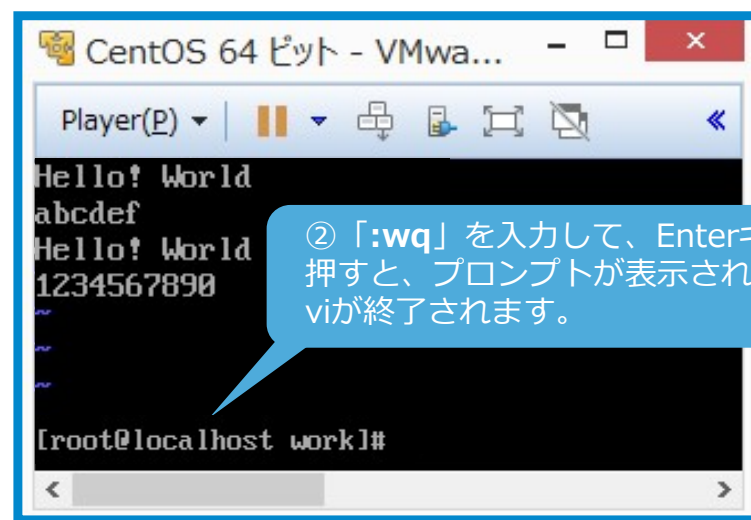
解説

切り取りと貼り付けの練習をします。以下のコマンドを実行して下さい。  
最後は「:wq」で保存してviエディタを終了しましょう。「w」コマンドと「q」コマンドを併せて使用します。



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
Hello! World
abcdef
Hello! World
1234567890
~
~
~
```

①「1234567890」が貼り付けられます。



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
Hello! World
abcdef
Hello! World
1234567890
~
~
~
[root@localhost work]#
```

②「:wq」を入力して、Enterキーを押すと、プロンプトが表示されて、viが終了されます。

# 文字列の検索①

## 解説

ファイル内に記述された文字を検索してみましょう。以下の手順に沿って操作してみてください。

※「」は半角スペースです。

順番	入力操作	意味	実行結果
1	<code>vi /etc/kdump.conf</code>	ファイル「kdump.conf」をviエディタで開きます	ファイル「kdump.conf」の内容が表示される
2	<code>/</code> キーを押す	コマンドモードが検索の状態になる	画面下に <code>/</code> が表示される
3	「device」と入力し、Enterキーを押す	ファイル「kdump.conf」上にある文字列「device」を検索する	カーソルが1件目の文字列「device」にジャンプする
4	<code>n</code> キーを数回押す	現在いる位置より下の検索結果にジャンプする	カーソルが次の文字列「device」にジャンプする
5	<code>N</code> キーを数回押す	現在いる位置より上の検索結果にジャンプする	カーソルが前の文字列「device」にジャンプする
6	<code>:q</code> を押して終了	ファイル「kdump.conf」を閉じてviエディタを終了します	プロンプトが表示される

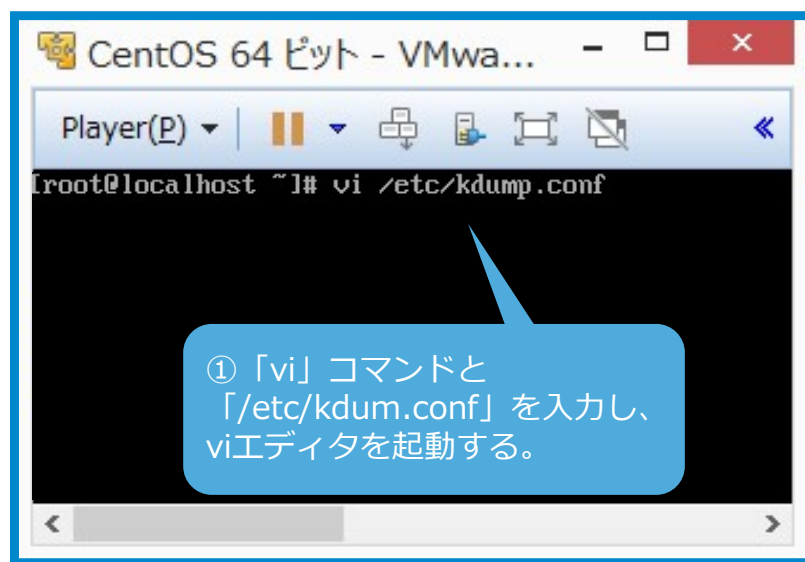


## 文字列の検索②

解説

入力手順に沿ってコマンドを打ってみましょう。


### 手順①



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
[root@localhost ~]# vi /etc/kdump.conf
```

① 「vi」 コマンドと「/etc/kdum.conf」を入力し、viエディタを起動する。

### 手順②



```
CentOS 64 ビット - VMwa...
Player(P) | [Icons]
# This file contains a series of commands to perform (in order) in the kdump
# kernel after a kernel crash in the crash kernel(1st kernel) has happened.
#
# Directives in this file are only applicable to the kdump initramfs, and have
# no effect once the root filesystem is mounted and the normal init scripts are
#
~~~~~
                        中盤省略
~~~~~
#
#proc/vmcore to <mnt>/<path>/%DATE%.
NOTE: <partition> can be a device node, label or uuid.
```

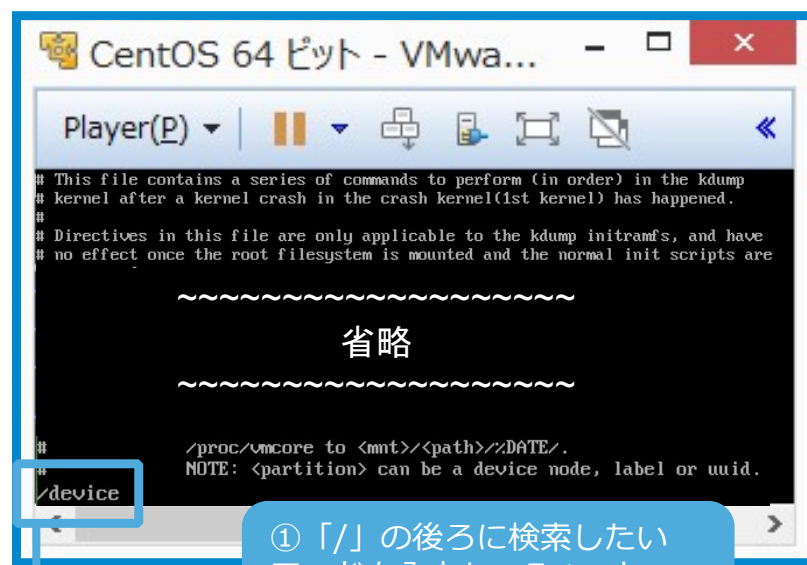
② 「/」を入力する。

# 文字列の検索③

解説

入力手順に沿ってコマンドを打ってみましょう。

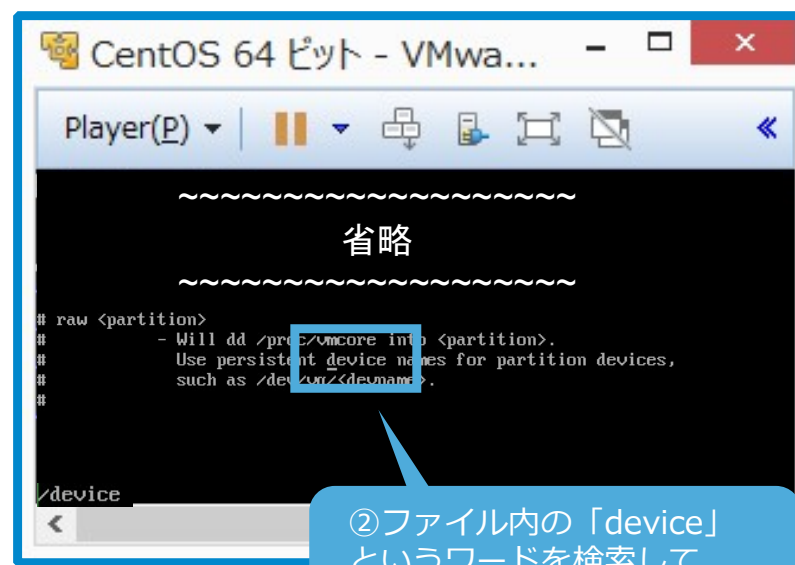
## 手順③



The screenshot shows a terminal window titled "CentOS 64 ビット - VMwa...". The terminal content includes instructions for searching for a word in a file. A blue box highlights the word "device" at the bottom of the terminal. A blue callout bubble points to the "device" word with the text: "①「/」の後ろに検索したいワードを入力し、Enterキー。今回は「device」とする。"

```
# This file contains a series of commands to perform (in order) in the kdump
# kernel after a kernel crash in the crash kernel(1st kernel) has happened.
#
# Directives in this file are only applicable to the kdump initramfs, and have
# no effect once the root filesystem is mounted and the normal init scripts are
#
~~~~~
省略
~~~~~
#
# /proc/vmcore to <mnt>/<path>/DATE/.
# NOTE: <partition> can be a device node, label or uuid.
#
/device
```

## 手順④



The screenshot shows the same terminal window as in step 3. The terminal content now shows the search results for the word "device". A blue box highlights the word "device" in the search results. A blue callout bubble points to the "device" word with the text: "②ファイル内の「device」というワードを検索して、そのワードを表示している。"

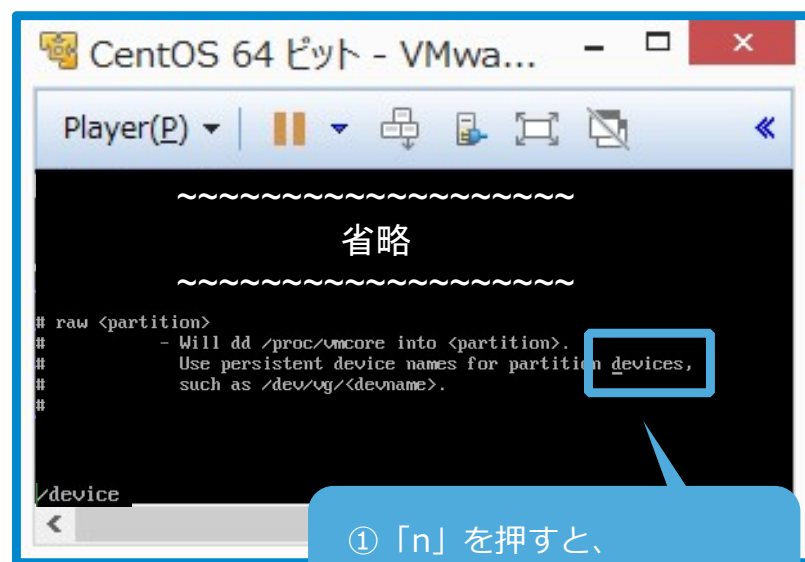
```
~~~~~
省略
~~~~~
# raw <partition>
# - Will dd /proc/vmcore into <partition>.
#   Use persistent device names for partition devices,
#   such as /dev/<devname>.
#
/device
```

## 文字列の検索④

解説

入力手順に沿ってコマンドを打ってみましょう。

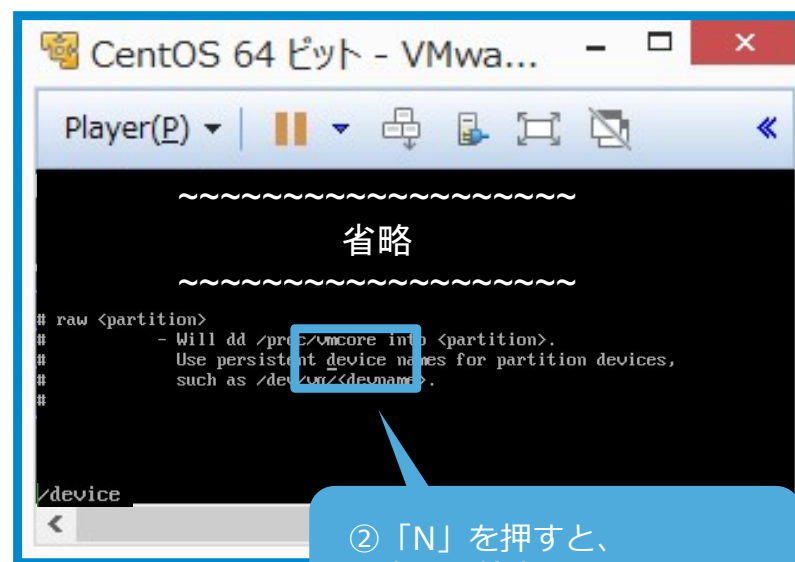
### 手順⑤



~~~~~  
省略  
~~~~~  
# raw <partition>  
# - Will dd /proc/vmcore into <partition>.  
# Use persistent device names for partition devices,  
# such as /dev/vg/<devname>.  
#  
#  
/device

①「n」を押すと、  
下方向に検索をかけていく。

### 手順⑥



~~~~~  
省略  
~~~~~  
# raw <partition>  
# - Will dd /proc/vmcore into <partition>.  
# Use persistent device names for partition devices,  
# such as /dev/vg/<devname>.  
#  
#  
/device

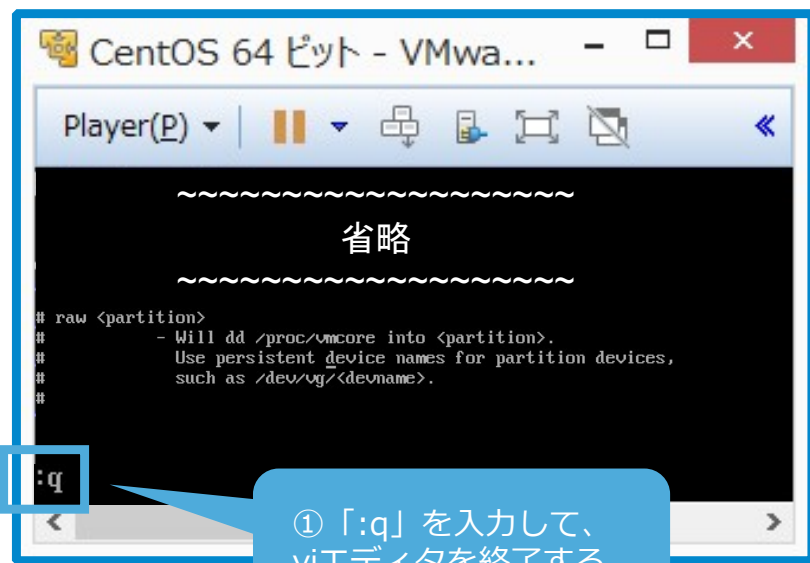
②「N」を押すと、  
上方向に検索をかけていく。

## 文字列の検索⑤

解説

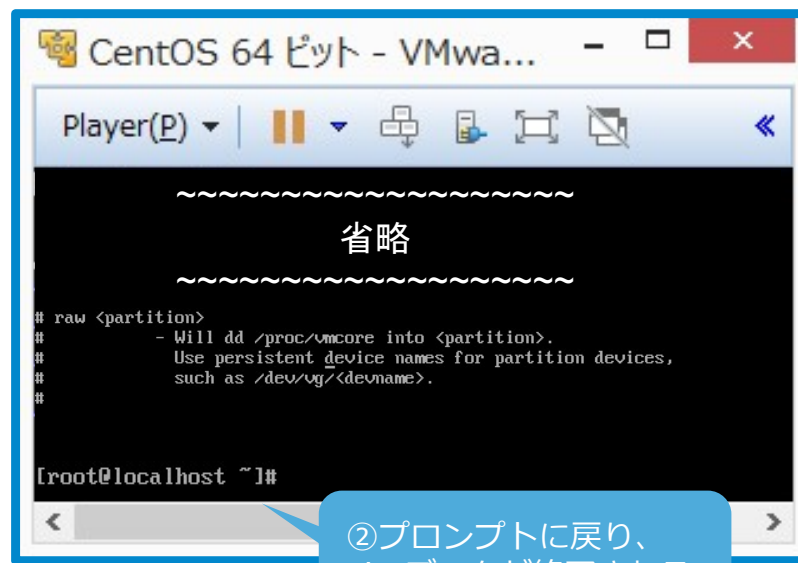
入力手順に沿ってコマンドを打ってみましょう。

## 手順⑦



①「:q」を入力して、  
viエディタを終了する。

## 手順⑧



②プロンプトに戻り、viエディタが終了される。