

Linux を知ろう！コマンドライン編



株式会社ALJ Education Plus (LPI-Japanアカデミック認定校)

廣出 啓太

LPI-JAPAN

■講師プロフィール



- ◆ 名前 : 廣出 啓太(ひろで けいた)
- ◆ 出身 : 愛知県出身
- ◆ 年齢 : 25歳
- ◆ 興味ある分野 : セキュリティ(主に暗号系)
- ◆ 使用言語 : Python, VBA, JavaScript, Ruby on Rails
- ◆ 使用環境 : Linux, Windows, Cisco, Nokia, AWS, MS Azure
- ◆ 趣味 : ライブ参戦(フジファブリックが好きです!)
バイクに乗ること
- ◆ 資格 : 応用情報技術者認定, CCNA, 危険物取扱者
- ◆ 経歴

2021年から新卒で入った会社でネットワークエンジニアとしてのキャリアを開始。
エンタープライズ、モバイルコアの設計構築案件に従事しながら
専門学校講師として学生にLinuxを教育。
またPython, VBAを使用した不動産会社のRPA開発を兼任。

最近、知り合いの会社のHP開発をしています！

■LinuCとは

クラウド／DX時代のITエンジニアに求められるシステム構築から運用管理に必要なスキルを証明する技術者認定です。

クラウド活用に役立つスキルの習得

- オンプレミス／仮想化・コンテナを問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解

習得できるスキルが実践的

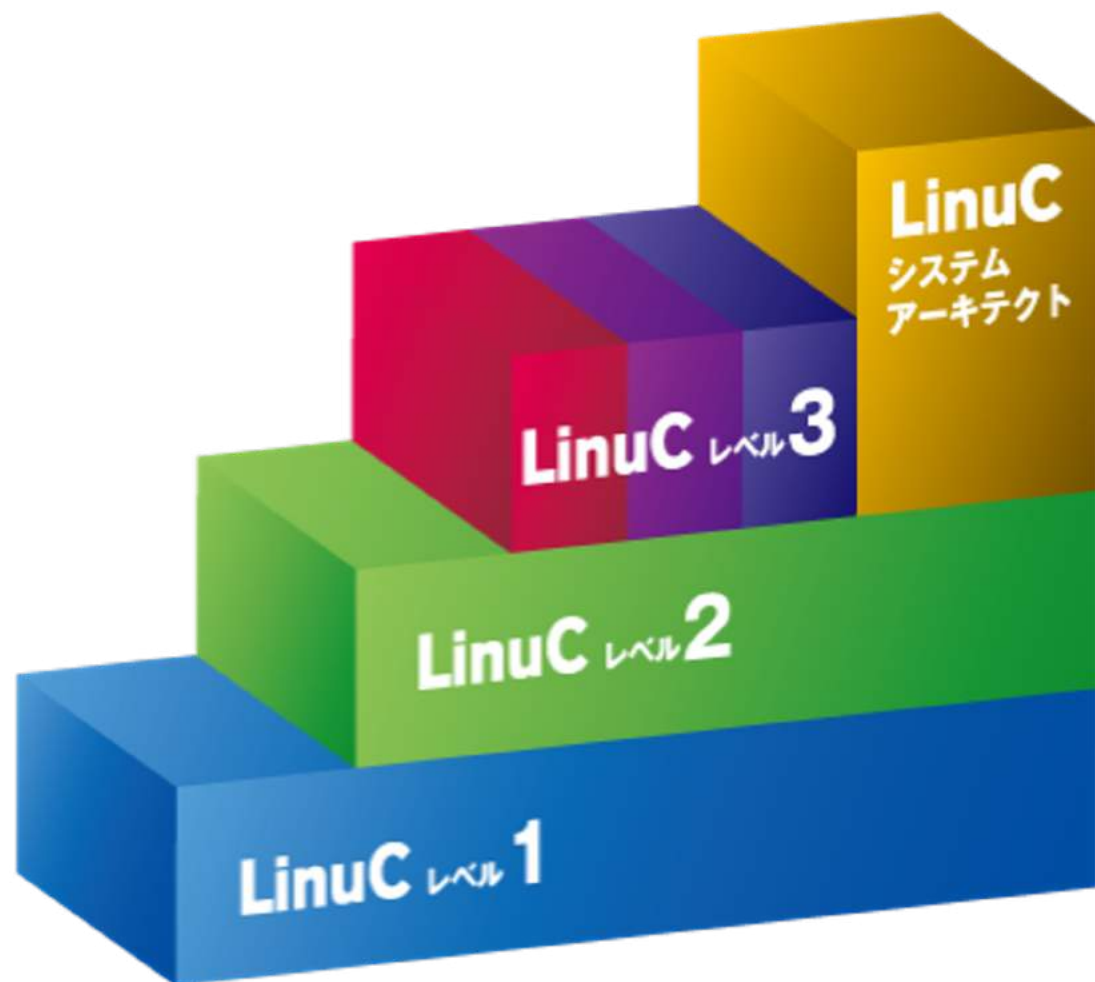
問題作成にはトップエンジニアも参加するコミュニティ内の意見を取り込むことで、本当に必要な内容を網羅的に盛り込んでいます。

上流工程を担うアーキテクトの領域までカバー

システムの運用管理からアーキテクチャ設計までの4つのレベルをひとつずつ習得していくことで、活躍できるエンジニアとして必要なスキルを網羅的に身につけていくことができます



LinuCは、サーバーの運用管理からアーキテクト設計まで、システム開発・運用に必要な知識とスキルを体系立てて習得することができます。



LinuC システムアーキテクト

ITプロジェクトを成功に導く上級エンジニア

SA01試験

SA02試験

LinuC レベル3

高度な技術力を備えた特定分野のスペシャリスト

304試験 (仮想化&高可用性)

300試験
(運用環境)

303試験
(セキュリティ)

LinuC レベル2

仮想マシン・コンテナを含むLinuxシステム、ネットワークの設計・構築

201試験

202試験

LinuC レベル1

物理/仮想Linuxサーバーの操作・運用

101試験

102試験

今回のテーマ

Linux を知ろう！コマンドライン編

導入～ Linuxについて

■Linuxとは

Linuxは、オープンソースの**オペレーティングシステム (OS)** です。

オペレーティングシステムとは、CPU、メモリ、ストレージなどのシステムのハードウェアとリソースを直接管理するソフトウェアのことで、アプリケーションとハードウェアの間に位置し、すべてのソフトウェアと作業を実行する物理リソースをつないでいます。

(RedHat. Linux とは.Linuxについて参照)

簡単に言うと・・・

皆さんが普段使ってる**Windows**や**MAC**などの仲間です



例として出すのであれば、普段みなさんが使用されてるPCも
 動画再生ボタンを押したら、画面が動いたり、音楽が流れたりしますよね
 いろいろ**物理的なものとの仲介屋**をして**くれているシステム**ということですよ

■ 普段身の回りに隠れているLinux

- Android(携帯)
- テレビ
- コピー機
- ネットワーク機器 など。



実は意識していないだけで**Linux**というOSは普段から身の回りに多数存在しています

なぜこんな身の回りにあるのか・・・

それはこのカンファレンスの名前にもある通り

Open Sourceなシステムであるからです

- じゃあ結局のところ、Open Sourceってなに？

このような質問が出てくるかと思えます

なんで**Open Source**について話し合うような場があるのか？

それは、**Linux**がここまで普及する理由になった1つの要因が含まれています

なぜか。

それは**Open**な**Source**だからです

何を言ってるのこの人って思った方次のページで説明します

Openなソースであること

これはLinuxOSの開発者がコードを公開しているということです

みなさんは普段、新しいジャンルのご飯を作るとき、レシピをみますよね？

美味しかったらリピートして行って
何回目かで、もしかしたらこの隠し味入れたらもっとおいしくなるのではないかと
思って作ったりしたことがあるかと思います



そしてそれが家庭の味になって、代々受け継がれていたり、新しい人に共有したり・・・

まさにこれがOpenなSource(レシピ)ということです

LinuxOSのソース(レシピ)が公開されていることでどんどん新しいものが開発されて行って
自分なりの成果物を作成していき・・・という形でLinuxがここまで普及していききました

Linuxコマンドについて
代表的なLinuxコマンドそれぞれの特徴

ある程度**Linux**についてどういうものか理解したかと思います

ではその**Linux**を実際にどうやっていじっていくか
そのための**コマンド**などを紹介していきます

今回は**Cent OS**を使用したレクチャーをしていきますが
基本的なコマンドは**他の種類のディストリビューション**でも使えるので
何を使用されても構いません

※Cent OSなどは**ディストリビューション**といい、開発者が作成した者を配布してくれています



■ ls (エルエス)

ディレクトリ(フォルダ)内の**ファイル**と**ディレクトリを表示する**コマンド

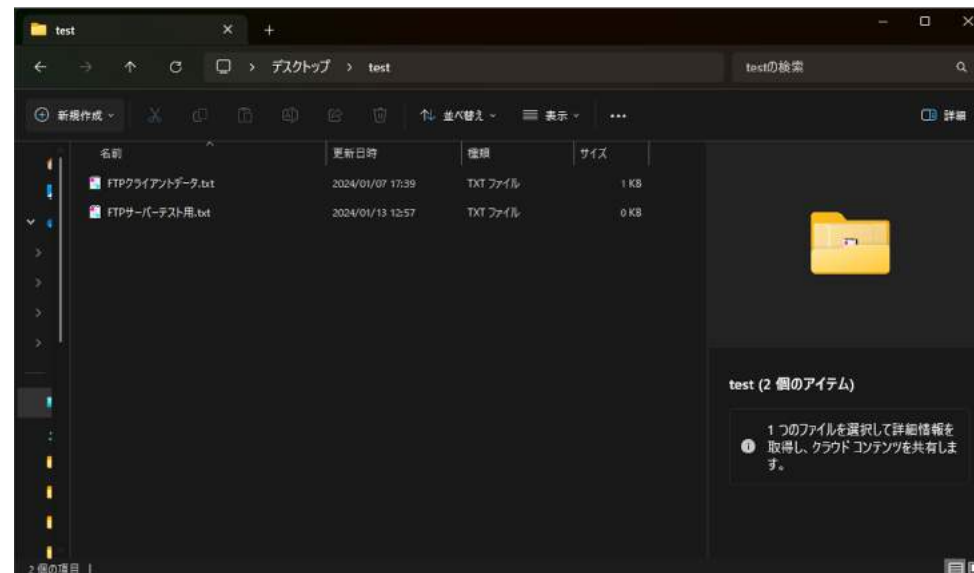
構文: **ls [表示したいディレクトリ]** ※何も入れない場合は現在の場所のファイルとディレクトリを表示

```
[hirode@localhost ~]$ ls
Mails            ダウンロード   デスクトップ   ビデオ   画像
conf_backup     テンプレート   ドキュメント   音楽     公開
```

普段エクスプローラーでこの**フォルダ**には何が入っているか確認しますよね

そのための**CLI**のコマンドです

※CLI: 黒い画面でコマンド入力したりするところ





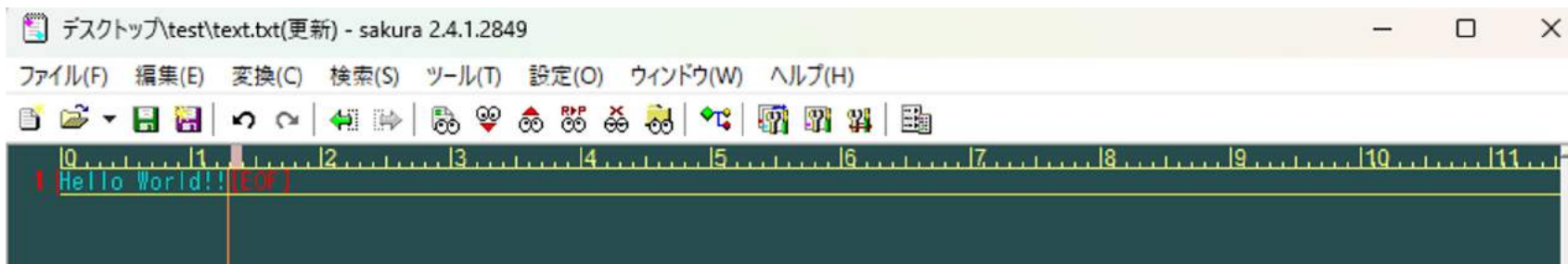
■ cat (キャット)

ファイルの**中身を確認**するコマンド

構文: cat **[確認したいファイル]**

```
[hirode@localhost ~]$ cat test.txt
Hello World!!
```

これも普段ダブルクリックで**メモ帳**で中身を見るのと同じです



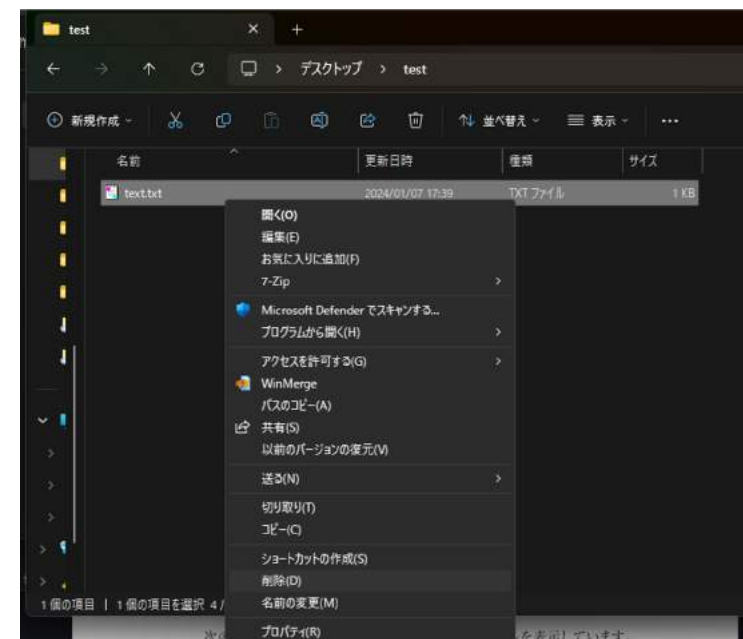
■ rm (アールエム)

ファイルなどを**削除**するためのコマンド

構文: rm **[削除したいファイル]**

```
[hirode@localhost デスクトップ]$ ls
test.txt
[hirode@localhost デスクトップ]$ rm test.txt
[hirode@localhost デスクトップ]$ ls
[hirode@localhost デスクトップ]$
```

ご存じの通り、**ファイルを消す**コマンドです





■ vi (ブイアイ)

ファイルなどを**編集**するためのエディタを起動するコマンド

※Viエディタと呼び、Linuxにデフォルトで存在していて、Windowsのメモ帳みたいなもの

構文: vi **[編集したいファイル]**

```
[hirode@localhost ~]$ vi test.txt
```

すこしだけややこしくなります。

vi には**インサートモード**、**コマンドモード**の2つがあります

viを実行し中身に入った後の最初のモードは**コマンドモード**です

編集するためのモードが**インサートモード**になるので
モードを変更する必要があります



■ vi (ブイアイ)

インサートモードに移行するためには **i** キーを押す必要があります
 移行すると以下のような表記になるかと思えます



こうなっている状態でいろいろ文字列を入力できるようになります



■ vi (ブイアイ)

文字列を入力して、保存したい場合に今まで**Ctrl + S**でできていた**保存**も**vi**ではそうはいきません
インサートモードを終了し、あるコマンドを入力することで**保存**できます
 それが **エスケープキー(Esc)**入力後 **:wq**です

まず**Esc**を押すことでインサートモードを終了し、コマンドモードへ戻ります
 その後呪文としての**コロン**、**w**(書き込み) **q**(終了)を意味し、書き込んで終了するという意味があります

では実際に保存してみましょう

このように入力することで
 ファイルを編集することができます

```
Hello World!!
```

```
:wq
```



■ pwd (ピーダブルディー)

カレントディレクトリ(今いる場所)を確認するコマンド

```
[hirode@localhost ~]$ pwd
/home/hirode
```

この出力結果の /home/hirode が今私が操作している場所です

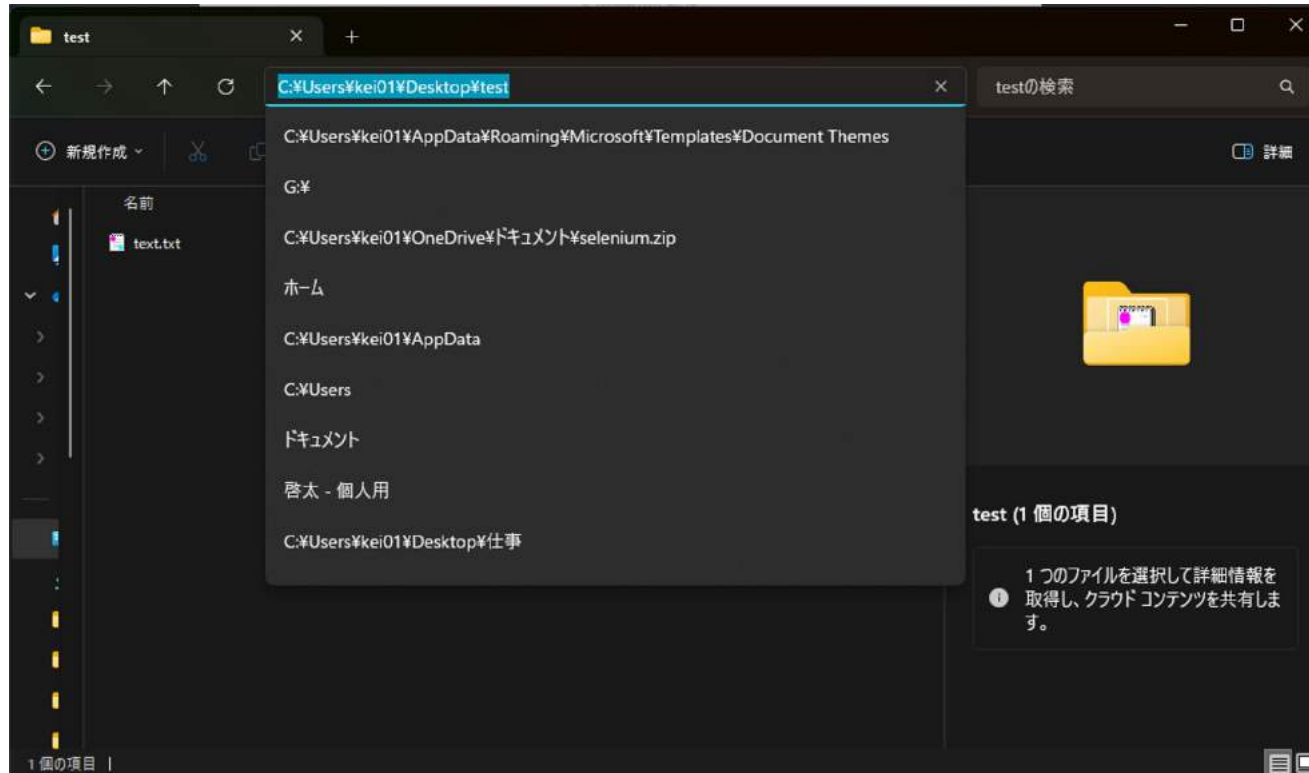
つまり。 / の下のhomeディレクトリのhirodeというディレクトリにいることを示してくれています

※この最初についている / という文字はルートディレクトリと言って一番上の階層にいることを示します



■ pwd (ピーダブルディー)

windowsでもこのようにリボンの場所にある個所をクリックすることで場所(PATH)を表示することができます





■ cd (シーディー)

カレントディレクトリ(今いる場所)を変更するコマンド

構文: cd [移動先]

```
[hirode@localhost ~]$ cd デスクトップ
[hirode@localhost デスクトップ]$
```

上記の例では"デスクトップ"に移動するコマンドになります

cdの何がよいのかというと本来であれば別ディレクトリにあるファイルなどは

cat デスクトップ/test.txt と対象のファイルが存在する**ディレクトリ**含め指定する必要がありました

先にデスクトップに移動することで、cat test.txtとパスを考えた**ディレクトリ**の指定が必要なくなります

※多くのファイルを編集したりするときに楽



■ cp (シーピー)

ファイルを**コピー**する為のコマンド

構文: cp [コピー元] [コピー先 or コピー後の名前]

```
[hirode@localhost ~]$ cp test.txt text2.txt
[hirode@localhost ~]$
```

cpはものすごく便利で**2種類**の使い道があります **※同時に使うことも可能**

- ① **コピー先**をディレクトリで指定することで、そのディレクトリにコピーする動作
- ② **コピー後の名前も指定**できる為、設定ファイルのバックアップなどに使用されます

(以下例) httpd.conf(Webサーバーの設定ファイル)を
conf_backup(適当に作成したディレクトリ)内に**コピーし名前を変更する**コマンド

```
[hirode@localhost ~]$ cp /etc/httpd/conf/httpd.conf conf_backup/bk_httpd.conf
[hirode@localhost ~]$
```



■ mv (エムヴィ)

ファイル or ディレクトリを①移動また②名前を変更する為のコマンド

構文: mv [移動元ファイル] [移動先 or 変更後の名前] ※同時に使うことも可能

①移動 (test.txtをtestdirディレクトリに移動)

```
[hirode@localhost ~]$ mv test.txt testdir
[hirode@localhost ~]$
```

②名前の変更 (test.txtをtest2.txtに変更)

```
[hirode@localhost ~]$ mv test.txt test2.txt
```

※cpと違い元ファイルを変更することになるので注意

Linuxコマンドについて
コマンドを使用してできることを紹介

今まで出てきたコマンドを使用すれば、基本的なことであればLinuxの操作が可能です
 以下のコマンドを習いましたが、このセクションではそれらを使用してできることを解説します

■ 出てきたコマンド

- ls
- cat
- rm
- vi
- pwd
- cd
- cp
- mv



ではこれらを使用し**実践的**に使用できることを紹介していきます

Case1

ちょっと君、このWEBサーバーのconfig変更しておいて！

■ Case1

ちょっと君、このWEBサーバーのconfig変更しておいて！

はい、商用環境でいきなりこんなことを言うてくる上司などは基本いないとは思いますがこの言葉には多くの意味が含まれています

1. configの変更依頼
2. 現状動いているconfigのbackupを取ること
3. 変更後、変更前のログも残すこと

①は言葉の通りですが②③はいきなりIT業界に入った人はわかんないですよね・・・
(もっと具体的に言ってほしいものですが、一応当たり前の動作ではあるので覚えておきましょう)

■ Case1

1. configの変更依頼
2. 現状動いているconfigのbackupを取ること
3. 変更後、変更前のログも残すこと

まず①についてですが、これは**基本コマンド**で編集自体はできます

viを使用することで、config自体は変更することができますが
中身をちゃんと編集するためにはそれぞれの設定値を知る必要があります

ここで②のトラップ発動！

いきなり**vi**で編集をせず、まずは現在動いているconfigファイルを**バックアップ**として保存しましょう

そのためには**cp**コマンドです(本来日付と時間を入れた名前にするのが望ましい)

```
hirode@localhost ~]$ cp /etc/httpd/conf/httpd.conf conf_backup/bk_httpd.conf
hirode@localhost ~]$
```

■ Case1

1. configの変更依頼
2. 現状動いているconfigのbackupを取ること
3. 変更後、変更前のログも残すこと

では②についてはクリアできたのでじゃあ実際に変更して行きましょう

はい、また**トラップ**③**発動**、オペレーションのログを残す為に変更前のログを取っておきましょう
 基本は使用しているterminalソフトの**ログ機能**を使用しているため、表示するだけで大丈夫です

そのためには **catコマンド**です

```
[hirode@localhost ~]$ cat /etc/httpd/conf/httpd.conf
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
```

■ Case1

1. configの変更依頼
2. 現状動いているconfigのbackupを取ること
3. **変更後、変更前のログも残すこと**

変更前のログも取得できたので、実際に変更していきます
 そのためのコマンドは前にも出てるとおり、viです

```
[hirode@localhost ~]$ vi /etc/httpd/conf/httpd.conf
```

viで編集後 :wqで保存し終了できたのでこれで終・・・

ではないですね！

最後に変更後のログも残しましょう
 catで表示して終了です

**このように1つの言葉には多くの意味が含んでいるので作業依頼があったら
 ちゃんと先輩に確認するようにしましょう**

Case2

ちょっと君、変更後のファイル、ローカルに持ってきていて！

■ Case2

ちょっと君、変更後のファイル、ローカルに持ってきて！

知るか！と言いたくなるところですが、ごく真っ当な依頼なのでちゃんと実行しましょう

実はこの作業も今まで習ったコマンドに**似たもの**を使用すれば可能です

scpコマンドです

cpコマンドの構文を思い出してみてください。

構文: cp **[コピー元] [コピー先 or コピー後の名前]**

scpコマンドも同じ構文で使用することができます

では次のページで実践して行きます

■ Case2

ちょっと君、変更後のファイル、ローカルに持ってきて！

コピー先は**通信できる**先であれば何でも大丈夫です！

※**通信できる** = **疎通性がある** といいます

つまり

scp [コピー元] [疎通性があるコピー先のIPアドレス:[ディレクトリ]]で実施することができます

```
[hirode@localhost ~]$ sudo scp /etc/httpd/conf/httpd.conf 192.168.1.2:/
root@192.168.1.2's password:
httpd.conf 100% 11KB 1.2MB/s 00:00
```

このように、**scpコマンド**を使用してローカルの**IPアドレス**を指定すればファイルを持ってくることができます

※**IPアドレス** = **住所のようなもの**

例として出すと

愛知県の名古屋城 を 東京に移転したい場合

[**scp** 愛知県/名古屋城 東京:]となるわけです

Linuxコマンドについて コマンドの特徴の覚え方

■ いきなり“今までのコマンド全部覚えてください！”

って言われても無理だと思います

ですが**基本的なコマンド**になるので、なるべく覚えておくべき！
ということでこのセクションでは覚え方について紹介していきます

特徴を覚えるために**現実世界**に例えて使用していきましょう

エンジニアジョークとしても使えるかと思います！



■ Case1

ちょっと今オフィスだから、今から居酒屋移動する！

これをコマンドにして覚えましょう！



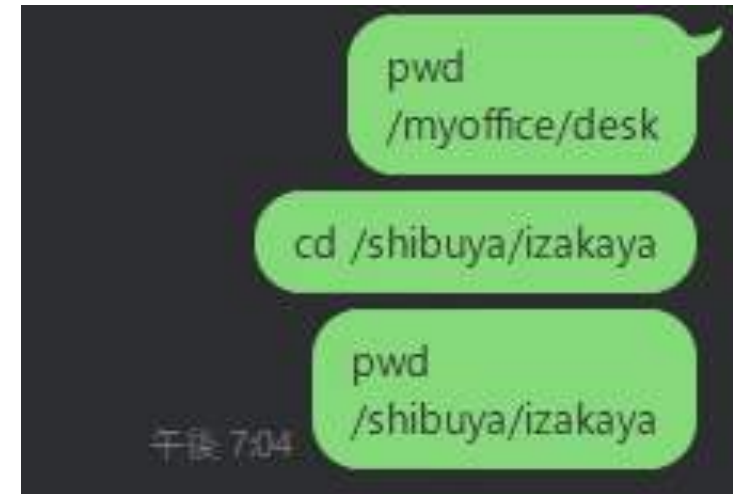
どうなるかというと・・・

```
pwd(現在地の表示)
/myoffice/desk
```

```
cd(実際の移動)
/Shibuya/izakaya
```

```
pwd(現在値の表示) ※要約:ついた！
/Shibuya/izakaya
```

これで渋谷の居酒屋に移動することができましたね！



■ Case2

(居酒屋にて)5000円現金ないわ。。Paypayで送るわ！

この場合どうなるでしょうか？

どうなるかというと・・・

```
mv /Paypay/5000yen /You/Paypay
```

mvの構文は

構文: mv [移動元ファイル] [移動先 or 変更後の名前] なので・・・

私のPaypayにある5000円をあなたのPaypayに**移動する**という意味になります

```
cp /Paypay/5000yen /You/Paypay
```

では絶対だめですね！

意味不明な5000円が**自分の場所にも存在したまま**なので、税務署に怒られます！

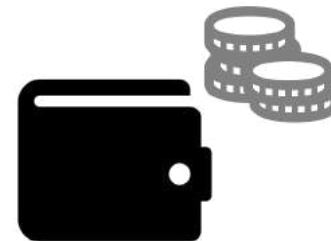
※もちろんこのコマンドでは5000円の送金できないのでちゃんと送みましょう



■ Case3

いま5000円送ったから、確認しておいて！

友人はこう言われたらどうやって確認すればよいでしょうか？



ls /You/Paypay(Paypayの中のファイルを確認)
(出力結果)5000yen

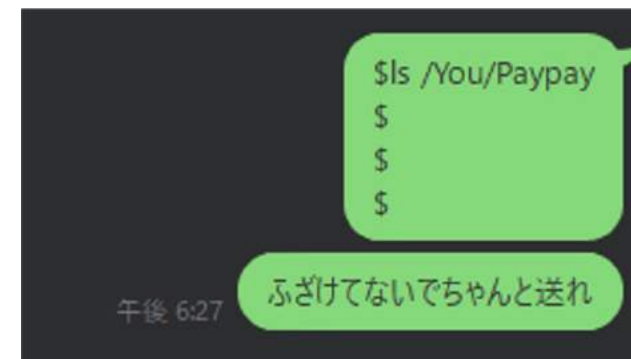
```
[hirode@localhost Paypay]$ ls
5000yen
```

こうなっていたらちゃんと送金できているのが確認できますね！
友人はcase2で送った場所を確認するようにしましょう

mv /Paypay/5000yen /You/Paypay

私は/You/Paypayに送っているのだから、それ以外の場所を見ても存在しないので、もめる原因になってしまいます！

まあこのように相手がエンジニアボケしたら付き合っただけあげましょう



Linuxコマンドについて

(応用)コマンドオプションを使用した便利術

■ では最後に応用編として

今までのコマンドがすこし便利になるような**オプション**を紹介します

LinuCの試験を受験するためには、**オプション**それぞれの意味を覚える必要があります

ですが、今回は一部の**オプション**を紹介し、実際の業務に使用できるかも？というものを抽出していきます

もっと便利なものも存在するので、これで興味がでたらぜひ**LinuCの参考書**などを読んでみてください！



[Linux教科書 LinuCレベル1 Version 10.0対応](#)

[中島 能和](#) (著), [濱野 賢一郎](#) (監修)

¥4,180



興味がある人は本をクリック！

■ ls -la

-l: 詳細な情報の表示

-a: すべてのファイルを表示

※optionの -l -aは連結して、-laとすることができます

【使用するタイミング】

Permission,ファイルの所有者,タイムスタンプ(いつ変更・作成したか)を確認したい時

※Permission: ファイルを編集したり、実行したりの権限

```
[hirode@localhost デスクトップ]$ ls -la
合計 4
drwxr-xr-x.  2 hirode hirode   56  2月  2 10:47 .
drwx----- 20 hirode hirode 4096  2月  2 10:44 ..
-rw-rw-r--  1 hirode hirode   0  2月  2 10:47 test.txt
-rw-rw-r--  1 hirode hirode   0  2月  2 10:47 test1.txt
-rw-rw-r--  1 hirode hirode   0  2月  2 10:47 test2.txt
[hirode@localhost デスクトップ]$
```

■ cat -n

-n:行番号の表示

【使用するタイミング】

configを確認するタイミング

行番号を確認しておくことで、相手にも変更場所を連絡しやすい

configファイルなどは結構な行数があります。そのため**行番号**を表示しないと大変なことになります

```
[hirode@localhost デスクトップ]$ cat -n /etc/httpd/conf/httpd.conf
 1 #
 2 # This is the main Apache HTTP server configuration file. It contains the
 3 # configuration directives that give the server its instructions.
 4 # See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
 5 # In particular, see
 6 # <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
 7 # for a discussion of each configuration directive.
 8 #
 9 # Do NOT simply read the instructions in here without understanding
10 # what they do. They're here only as hints or reminders. If you are unsure
11 # consult the online docs. You have been warned.
```

■ rm -rf

-r: 再帰的に削除

-f: 確認なしに削除

【使用するタイミング】

※注意: なんでも一気に消してしまいます。このコマンドを使用するときは良く確認してから使用しましょう。

ディレクトリなどは本来rmコマンドだけでは消すことができません

中に多くのファイルが存在している可能性があるため仕様としてできないようになっています

この**再帰的に消す**というのは、**中のファイルが消えるまでループして消す**といったイメージです

```
[hirode@localhost デスクトップ]$ rm test
rm: `test' を削除できません: ディレクトリです
[hirode@localhost デスクトップ]$ rm -rf test
[hirode@localhost デスクトップ]$ █
```

■ cp -ri

-r: 再帰的にコピー

-i: 同じファイルがあった場合に上書きするか確認する

【使用するタイミング】

ディレクトリごとコピーしたい時

※cp コマンドだけではディレクトリはコピーすることができません

以下表示のように同じ名前のファイルがあったときに上書きするかの確認ができるので更新したくないファイルが存在する場合は“n”と答えることで**スキップ**することができます

※-iではなく-fにすることでこういった確認も無視することができます

```
[hirode@localhost デスクトップ]$ cp test test1
cp: ディレクトリ `test' を省略しています
[hirode@localhost デスクトップ]$ cp -ri test test1
cp: `test1/test/test.txt' を上書きしますか? y
cp: `test1/test/test1.txt' を上書きしますか? y
cp: `test1/test/test2.txt' を上書きしますか? y
```

Linuxコマンドについて 最後に

■ いかがでしたでしょうか？

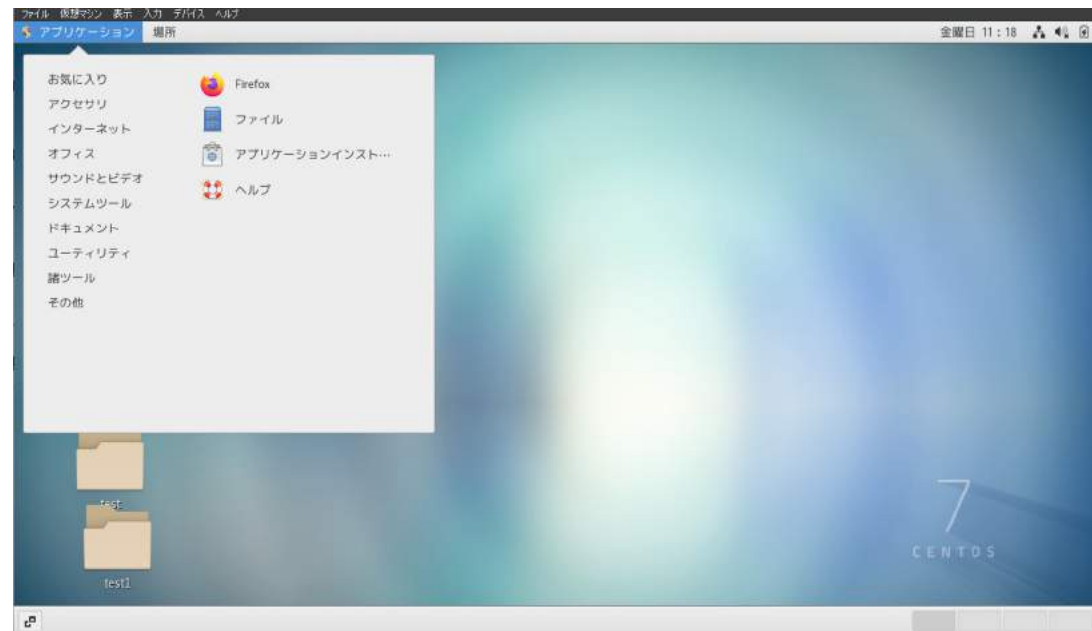
Linuxについて苦手知識を持っている人もいるかと思います

ですが**実際はPCの使い方講座**に近い内容になっています

難しく感じるのは黒い画面でパチパチとコマンドを打っていくことだけかと思います

ですが**Linux**にもマウスでクリックしたりと
いつも使用しているPCのように使用することもできます

もしこの講座で**Linux**に興味があれば
実際にインストールして試してみてください！



ご静聴ありがとうございました！