FreeBSD をさわってみよう

FreeBSD は、コマンド入力が基本です。コマンドはプロンプトが表示されている時だけ入力ができます。

```
% ■
```

■　がプロンプトです。

このときに、FreeBSD に対しておかしなコマンドを入力したりすると、プロンプトが表示されない事があります。表示されないと、コマンド入力を受け付けてもらえなくなります。そのような状態になってしまったら、次のような処置をしてみましょう。

Ctrl＋C　（コントロールキーを押しながら c を押す）
Ctrl＋D　（コントロールキーを押しながら d を押す）
Ctrl＋Q　（コントロールキーを押しながら q を押す）

を実行してみてください。だいたいプロンプトが表示されようになります。

どうにも、こうにも　プロンプトが表示されずにコマンドを受け付けてもらえない時は

１・オペレーションは駄目でも、キーボードがいきている時
　Ctrl＋Alt＋Del　を同時に押すと、再起動します。
　NumLock キーでキーボードのランプの反応を確認する。

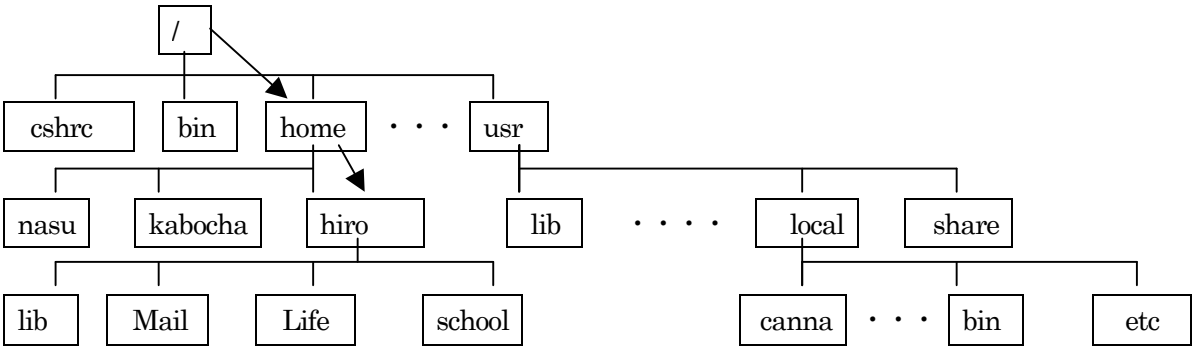
２・telnet 等でログインして、停止、再起動をする。

３・それでもだめで、キーボードの反応すらない時は、リセットボタンを押すか？電源を切るかです。

３でシステムを停止・再起動をしたときは、ファイルシステムにクリーンフラグを立てる事ができなくなることがあるために、次の起動の時に、fsck(8)のチェックを受ける事になります。

pwd

FreeBSD のファイルシステムは階層構造を用いています。
階層構造とは、下記図のように木の構造で、木の根にあたるものが　/　です。
/ から　ディレクトリ　と呼ばれる枝にあたるものがあります。
ディレクトリ　とは、ファイルを入れておく入れ物です。その中に、さらにファイルを入れておく事ができます。　　Windows で使用している「フォルダ」と呼ばれているものに、あたいします。



例えば上記の図のように
/ と言うファイルシステムの根元から　bin, usr, , home と言うディレクトリと、.cshrc や.profile と 言うファイルがあります。

FreeBSD で作業をしている時は、ユーザはどこかのディレクトリにいます。
自分が現在作業をしているディレクトリの事をカレントディレクトリといいます。

現在作業をしているディレクトリの事（カレントディレクトリ）を調べるには、
pwd と言うコマンドを実行します。

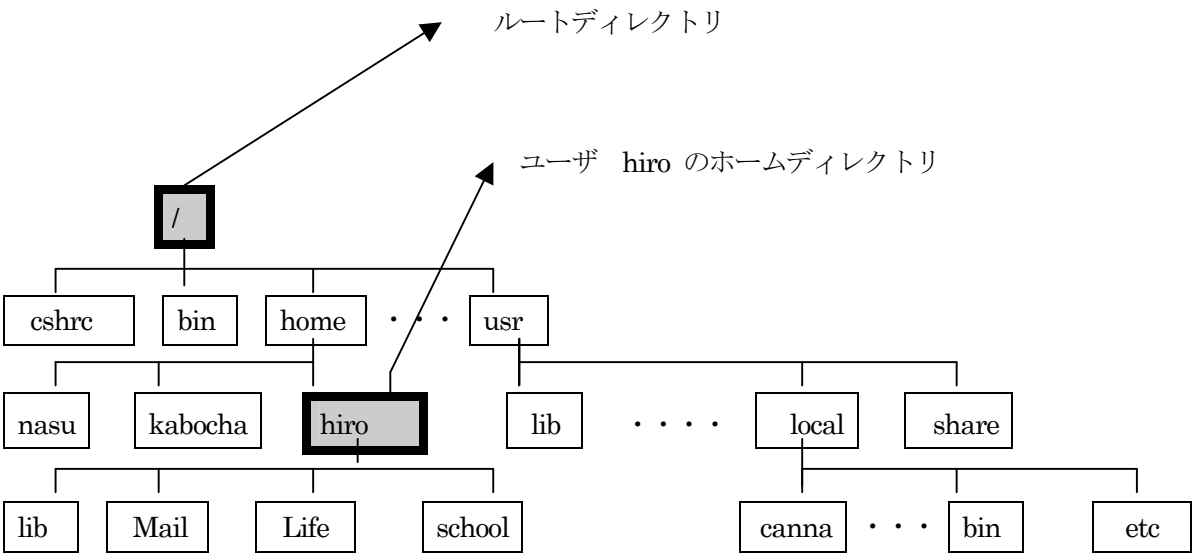
コマンドは　小文字、大文字の区別があるので、注意してください

```
% pwd
/home/hiro
%
```

/usr/home/hiro が　カレントディレクトリであることが解りました。

ファイルとファイルディレクトリ

ユーザ　hiro からみた　ツリー構造(木構造)



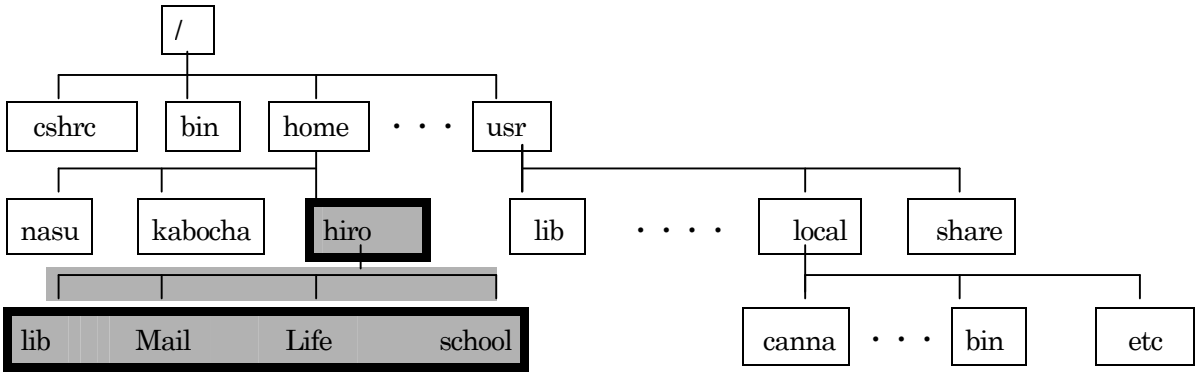
ルートディレクトリとは

ルートディレクトリ：ツリー型ディレクトリ構造の最上層にあるディレクトリのこと
コンピュータ上のすべてのファイルは必ずどこかのディレクトリに所属し、ルートディレクトリ以外のディレクトリは必ず他のディレクトリに所属している。
このような構造を木の枝の分かれ具合に例えた表現
が[ツリー（tree = 木）型ディレクトリ構造]
そして、このツリー構造の一番上にあるディレクトリを、木の根に例えた「ルート(root)」ディレクトリである(root は英語で「根」)
ちょうど上記の図をひっくり返して、ルートを下にすれば木が生えた形になります。

ホームディレクトリでは、ファイルやディレクトリの作成、削除、移動などの操作を自由にできる。
通常の作業は、ホームディレクトリと、ホームディレクトリより下の階層にあるディレクトリで、行う

ホームディレクトリ：ユーザごとに用意された、各ユーザが自由に利用できるディレクトリ。
ホームディレクトリが設定されたシステムでは、ユーザがログインすると、そのユーザのホームディレクトリに簡単にアクセスできるように設定されている。

ユーザ　hiro が　通常作業を行う領域(下記図のグレーの部分)



ユーザが直接扱う事が出来るファイルやディレクトリは、カレントディレクトリにあるものだけです。
しかし、カレントディレクトリ以外のファイルやディレクトリを扱うには、

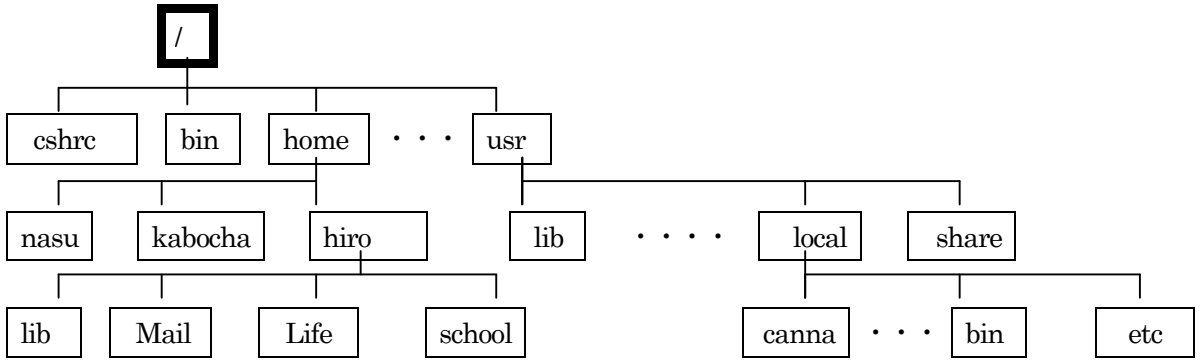
- ファイルやディレクトリのあるディレクトリに移ってから作業をする。
- 扱うファイルやディレクトリを、ツリー構造上の位置も含めて指定して作業をする。

対象になるファイルやディレクトリがある場所を指定する必要があります。
ツリー構造上の場所を指定するための経路の事をパス（path）と呼びます。
パスには、絶対パス　と　相対パス　が、あります。

絶対パス

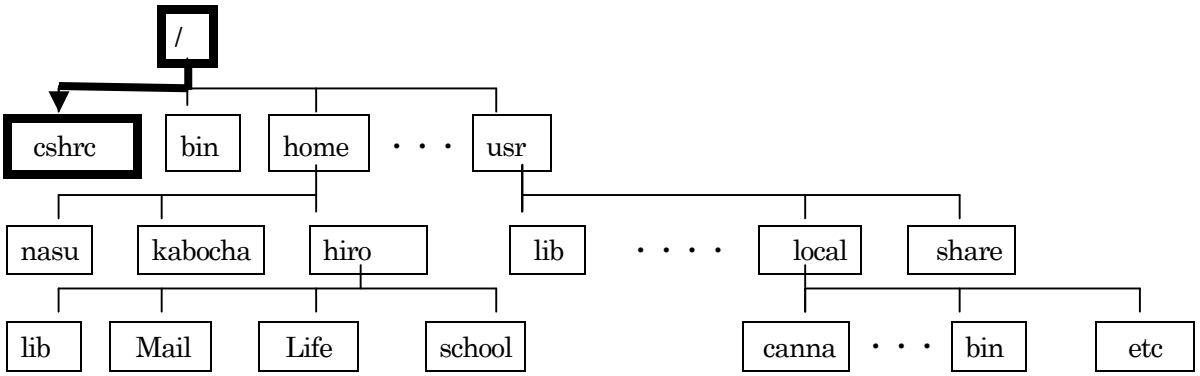
絶対パスは、ルートディレクトリから、目的となるディレクトリへの経路をあらわします。

まずは、/ から　/ までの絶対パスの位置をあらわします。



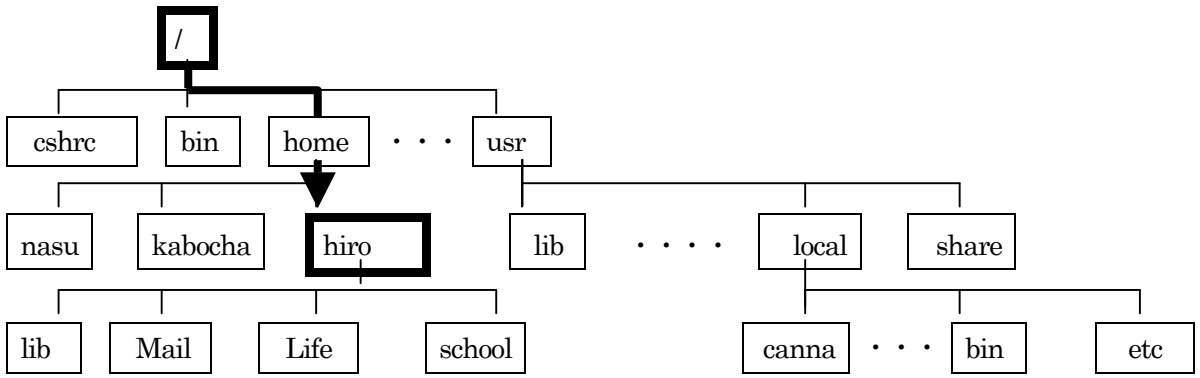
/　です。

/　から.cshrc　までの絶対パスの位置を表します。



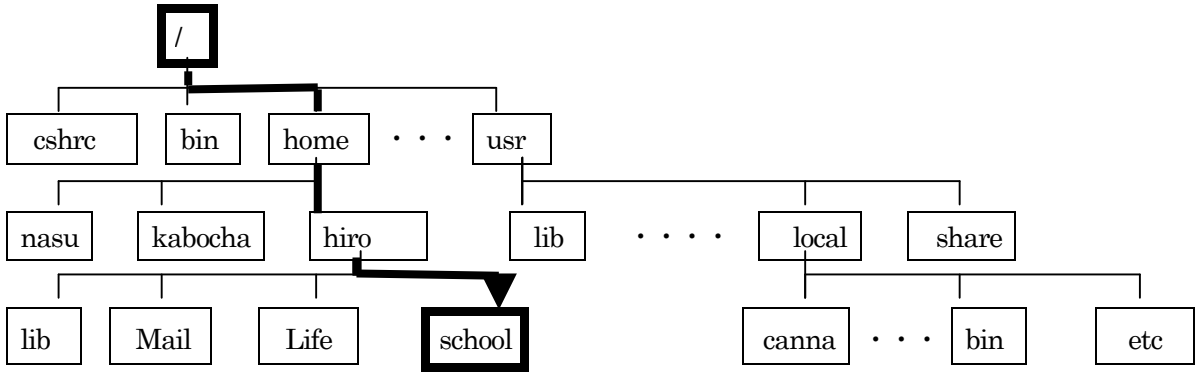
/.cshrc　です。

次は、/ から ホームディレクトリ　hiro までの絶対パスの位置を表します。



/home/hiro　　です。

次は　/ から、school までの　絶対パスの位置を　表します。



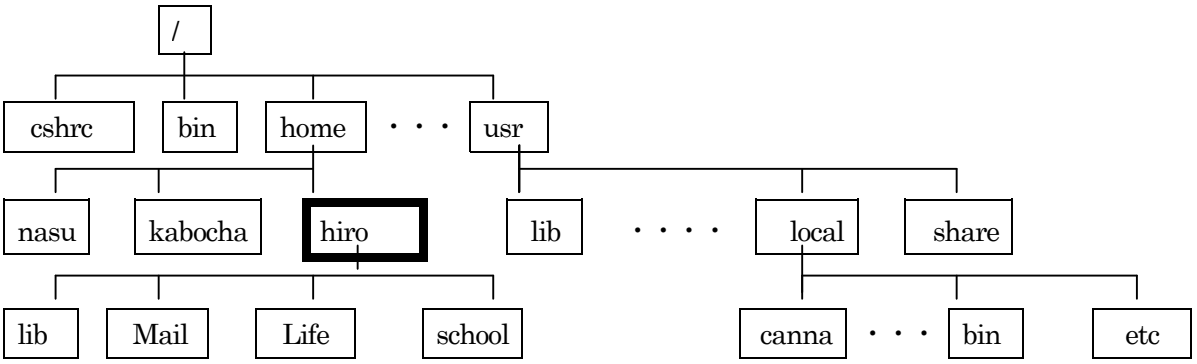
/home/hiro/school　　です。

相対パス

相対パスは、現在の自分の位置、カレントディレクトリから目的のディレクトリへの経路を表します。

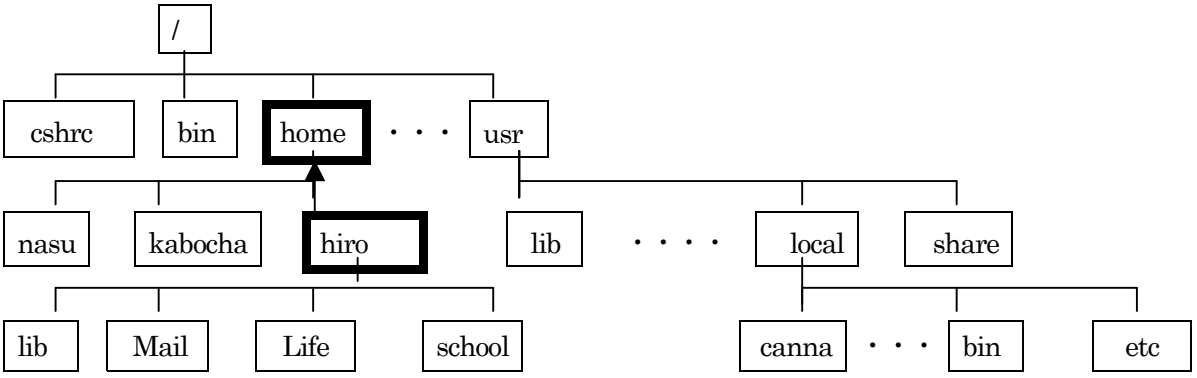
カレントディレクトリが異なれば目的のディレクトリは、同じであっても、パスは違う表し方をします。

hiro から　hiro　までの　相対パスです。



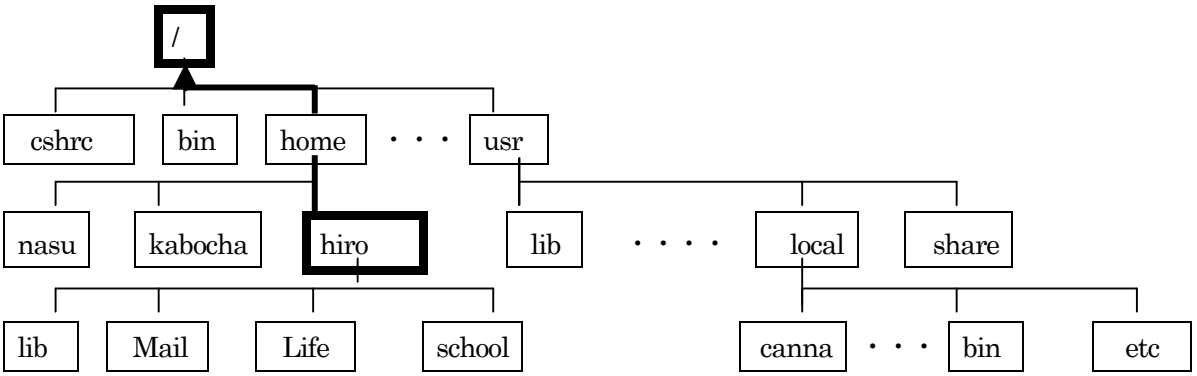
./ です。

hiro から　home までの　相対パスです。



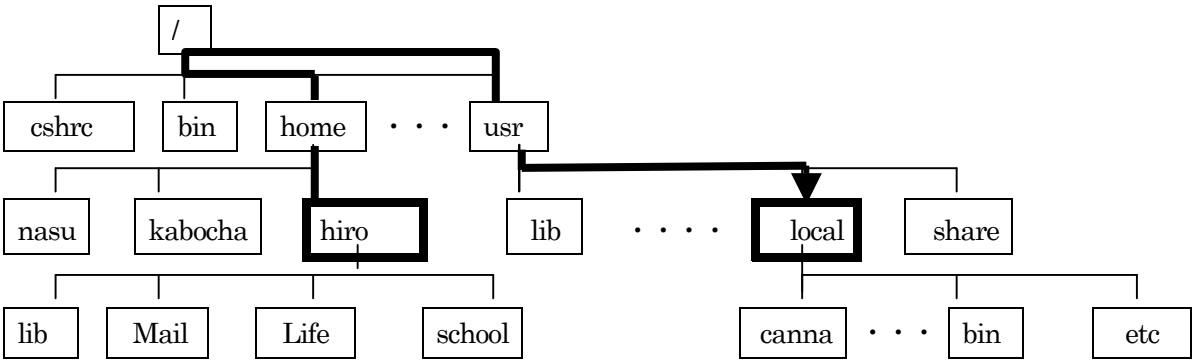
./../ です。

hiro　から　/ までの　相対パスです。



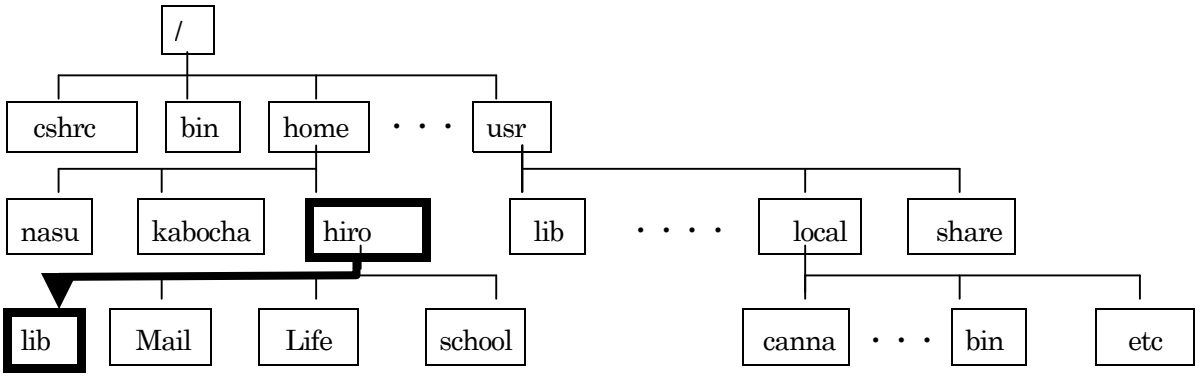
./../../　です。

hiro から　local までの相対パスです。



./../../usr/local　です。

hiro から　lib までの　相対パスです。



./lib です。

相対パスによる経路のを表すには、 “.“ や “..“ などのドット記号を使用します。

“./“ は、入力せずに省いてもよい。

dirctory の指定する特別な文字

.	現在いるディレクトリを指している
..	一つ上の階層のディレクトリを指している（親ディレクトリ）
~	ホームディレクトリをさしている

cd コマンドを引数なしで、実行した時は、環境変数 HOME で指定されたディレクトリに移動する。
環境変数 HOME で ホームディレクトリを設定しているから、ホームディレクトリに戻ります。

mkdir [オプション] ディレクトリｎ

使用例

ディレクトリ ffdir をつくります。
% mkdir ffdir

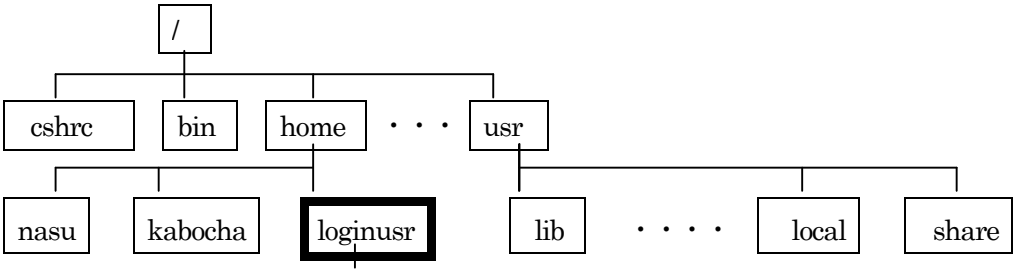
/usr/home/user に、ディレクトリ ffdir1, ffdir2, ffdir3 をつくります。

% mkdir /usr/home/user/ffdir1 /usr/home/user/ffdir2 /usr/home/user/ffdir3

ディレクトリをさわる。

ディレクトリとは、テキストファイルや、実行ファイル、　等　いろいろなファイルを入れる入れ物の名前です。

windows などで、言われている。ホルダーとおなじと考えてください。



ログインすると　まずホームディレクトリにいきます。
ホームディレクトリとは、ログインした時に、最初に訪れるディレクトリです。

ここでは、loginuser　がホームディレクトリとします。

pwd と言うコマンドを使ってまいりましょう。
pwd というコマンドは、現在いるディレクトリを表示します。
ここでは、/home/loginusr です。

この図の一番上の　/ だけを注目してください。　/ の事を、ルートディレクトリと呼びます。
次に下の階層に　home に注目してください。　home のディレクトリの位置をあらわすのに　/home と表現します。
次にもう一つ下の階層に　loginusr に注目してください。
loginusr のディレクトリの位置をあらわすのに　/home/loginusr と表現します。

どのようにあらわすかといいますと。下記の順になります。

```
/ のなかの   home/ のなかの   loginusr/       と表現すると
                        ↓   ↓
/              home/              loginusr/
                        ↓   ↓
    /home/loginusr/
```

まず、pwd と文字を打って　Enter キーを押します。

```
%pwd
/home/loginusr
%
```

ホームディレクトリが　/home/loginusr であることを確認します。
loginusr の文字は、読者のみなさまのログインユーザ名であることを確認ねがいます。

次に　ls コマンドで、ディレクトリの中身を確認します。

ls　と文字を打って　Enter キーを押して　ディレクトリ/home/loginusr の中を確認しましょう。
多分　なにも表示されていないと思います。
本当は、隠れていて見えないのですが、ここでは、見ないでください。

```
% ls
%
```

次にディレクトリを作成します。

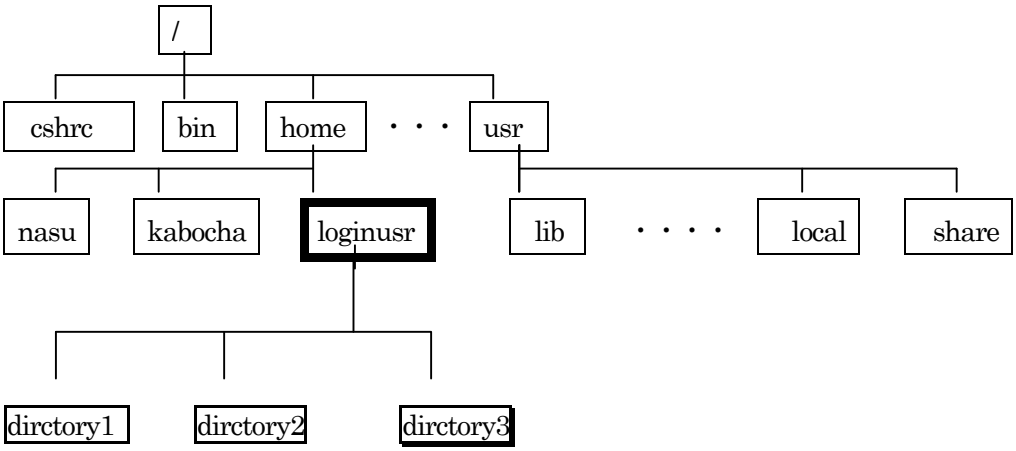
mkdir　と文字を打って　ファイルの名前を打ちます。次に Enter キーを押します。

（ディレクトリの名前は　dirctory1　dirctory2　dirctory3　の３つを作りましょう。）

次に、ls とコマンドを打って　/home/loginusr のディレクトリの中身を見てみます。

```
% mkdir dirctory1
% mkdir directory2
% mkdir dirctory3
% ls
    directory1    dirctory2    directory3
%
```

下記の図のように　ディレクトリ　　directory1　dirctory2　dirctory3 ができました。



次に、ディレクトリを削除してみましょう。

directory2 を削除してみましょう。

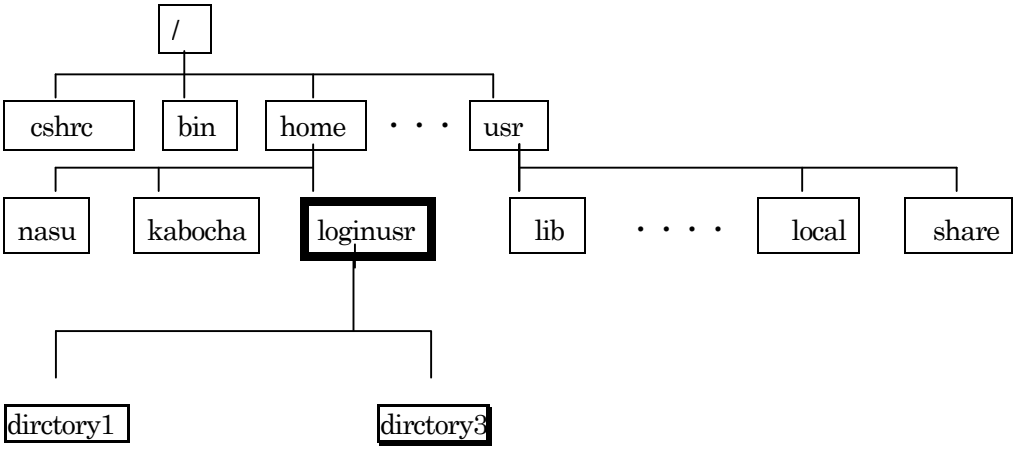
rmdir　コマンドをつかいます。

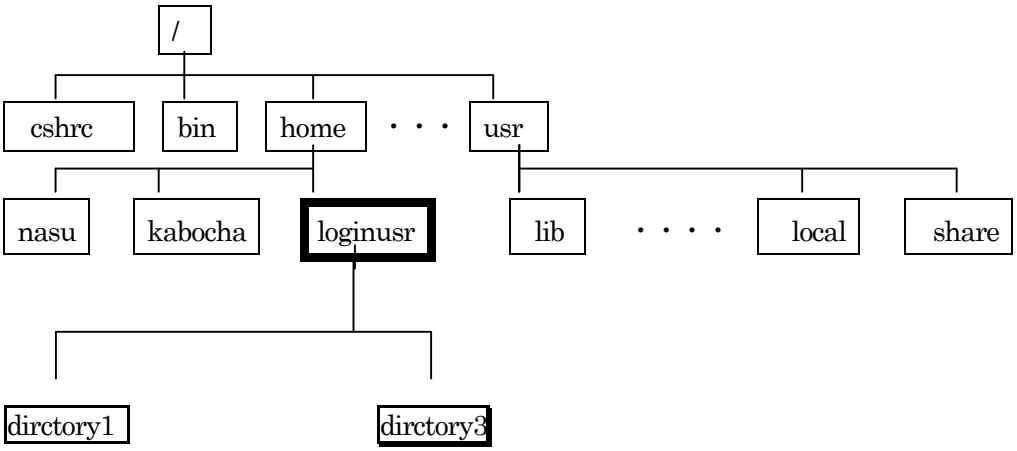
rmdir　ディレクトリ名

ここでは、
rm　dir　と文字を打ち込み　Enter キーを押します。

そして、ls コマンドで　directory2 が削除された事を確認します。

```
% rmdir directory2
% ls
   directory1          directory3
%
```





カレントディレクトリ

カレントディレクトリは、現在作業している場所です。
pwd というコマンドで確認できます。
例えば作業している場所がホームディレクトリならば
カレントディレクトリは、/home/hiro/loginusr です。

次は、カレントディレクトリを変更します。

現在のカレントディレクトリは、ホームディレクトリの　/home/hiro/loginusr　です。
ここから、一つ階層の下の dirctory 1 に移動してみます。

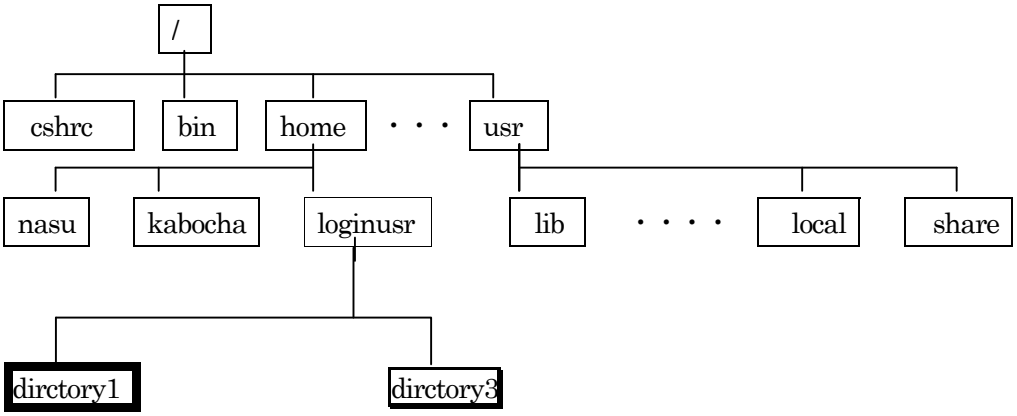
移動するには、cd コマンドをつかいます。(change directory)

cd ディレクトリ名

まず、cd dirctory1 と文字を打ち　Enter キーを押す。

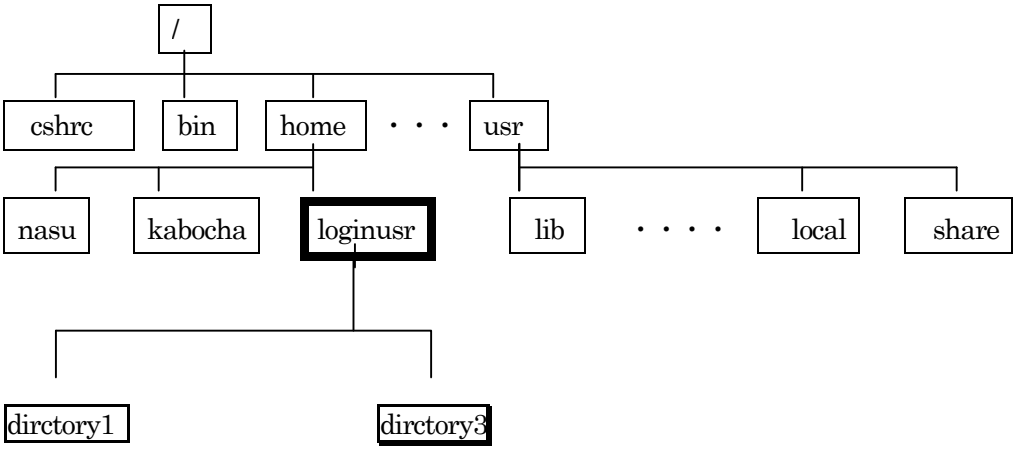
つぎに、pwd コマンドで、現在作業している。カレントディレクトリを表示させます。

```
% cd dirctory1
% pwd
  /home/hiro/dirctory
%
```



次にカレントディレクトリを/home/hiro/loginusr/dirctory1 から /home/hiro/loginusr/ もどります。

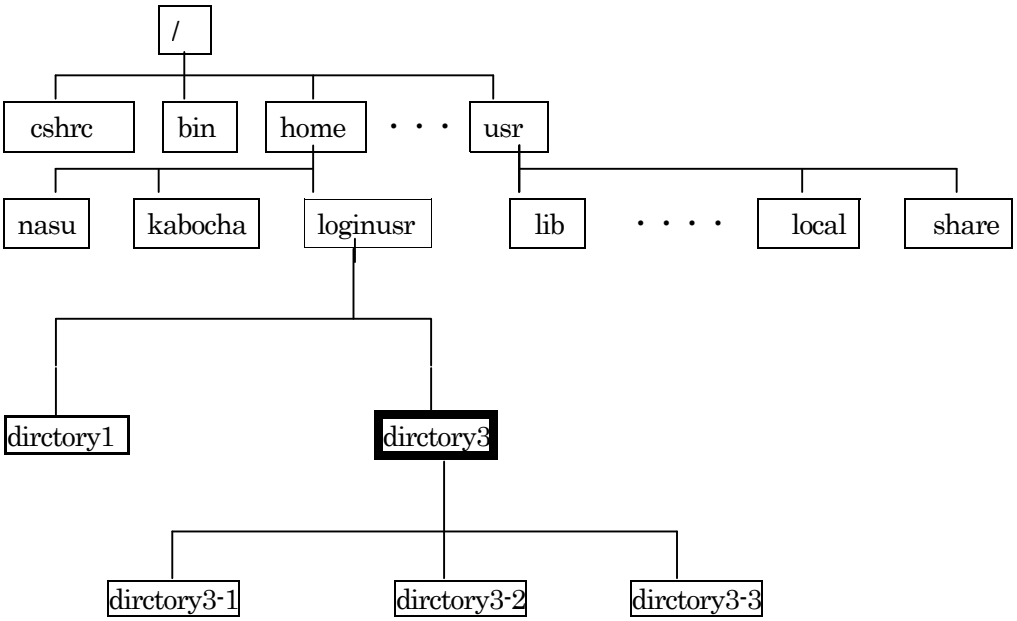
```
% cd /home/hiro/
% pwd
  /home/hiro
%
```



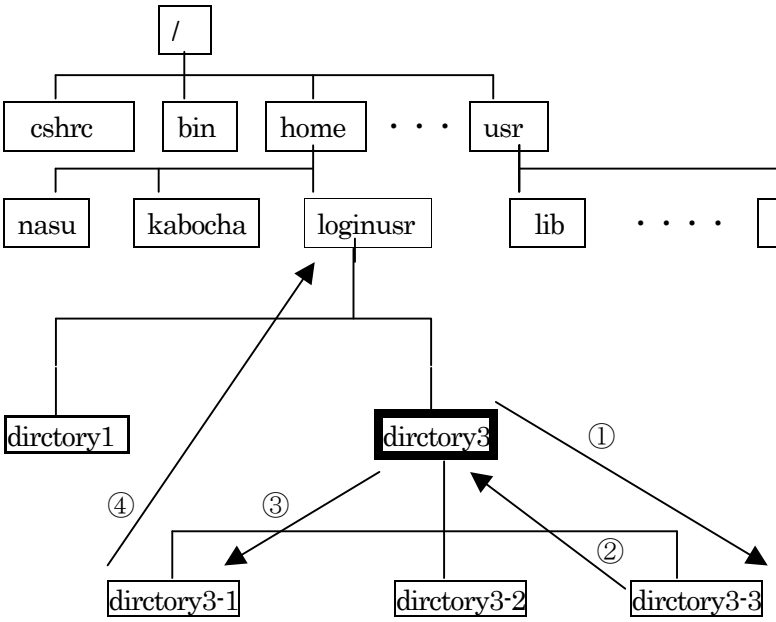
次は、/home/hiro/dirctory3　にカレントディレクトリを変更して、mkdir コマンドを使い新たにディレクトリをつくります。

ディレクトリ名は、　dirctory3-1　dirctory3-2　　dirctory3-3

```
% pwd
   /home/hiro
% cd /home/hiro/dirctory3
% pwd
   /home/hiro/dirctory3
% mkdir dirctory3-1
% mkdir dirctory3-2
% mkdir dirctory3-3
% ls
   dirctory3-1      dirctory3-2      dirctory3-3
%
```



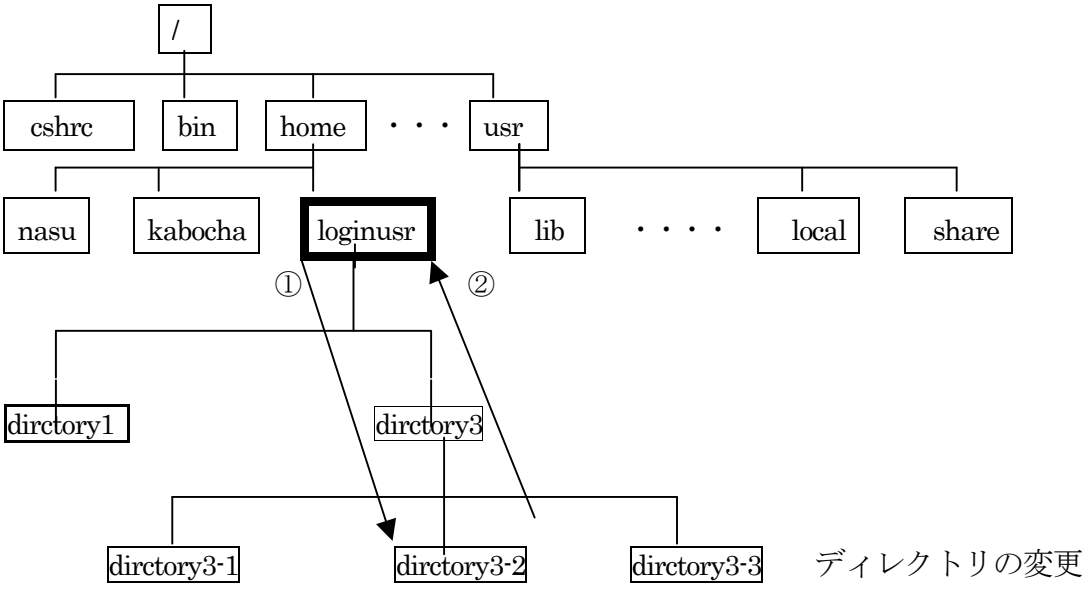
次は　カレントディレクトリを　/home/hiro/loginusr/dirctory3/dirctory3-3 に変更します。
pwd でカレントディレクトリを確認します。また一つ上の階層の/home/hiro/loginusr/dirctory3 に
戻るために、　cd　/home/hiro/loginusr/dirctory3 とはせずに、
cd ..　とコマンドを打つと　一つ上の階層に変更します。
pwd でカレントディレクトリを確認します。
次ぎに、カレントディレクトリを　/home/hiro/loginusr/dirctory3/dirctory3-1 に変更します。
pwd でカレントディレクトリを確認します。
最後のカレントディレクトリの変更は　cd /home/loginusr とせずに、cd とコマンドを打ちます。
cd とコマンドを打つと、カレントディレクトリがどこでも、ホームディレクトリにもどります。



カレントディレクトリの変更

```
%pwd
 /home/loginusr/dirctory3
% cd   /home/loginusr/dirctory3/dirctory3-3
% pwd
    /home/loginusr/dirctory3/dirctory3-3
% cd ..
% pwd
  /home/loginusr/dirctory3
```

```
% cd /home/loginusr/dirctory3/dirctory3-1
% pwd
  /home/loginusr/dirctory3/dirctory3-1
% cd
% pwd
  /home/loginusr
```



ディレクトリの変更

今までホームディレクトリから　/home/loginusr/dirctory3/dirctory3-2 のディレクトリに変更するのには、
cd /home/loginusr/dirctory3/dirctory3-2　とコマンドを打っていましたが、ホームディレクトリを
基準に
cd dirctory3/dirctory3-2 で変更できます。
/home/loginusr/ までを、省略できます。

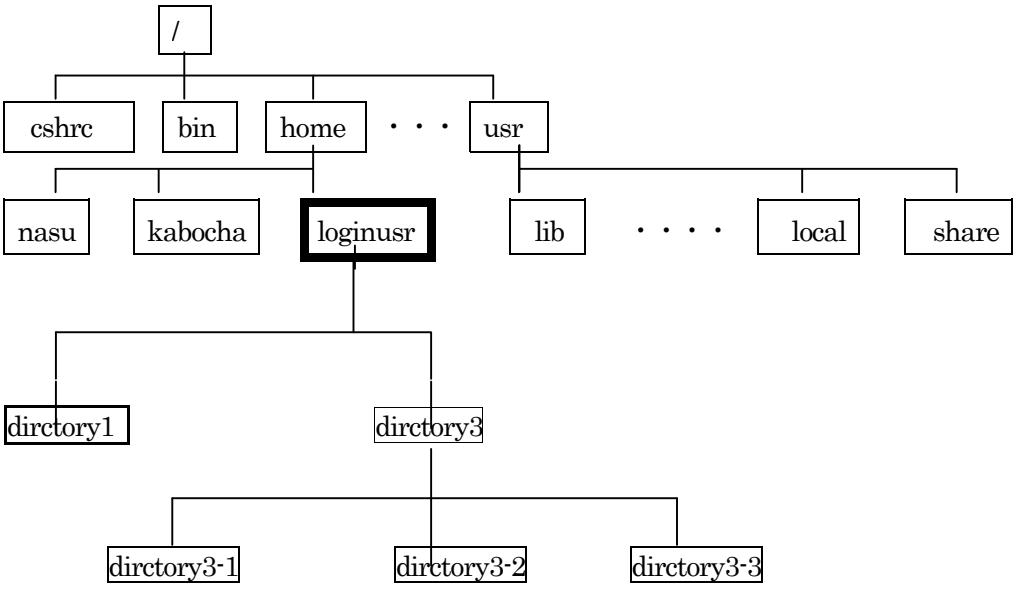
```
% cd dirctory3/dirctory3-2
% pwd
  /home/loginusr/dirctory3/dirctory3-2
% cd
% pwd
  /home/loginusr
%
```

このように、カレントディレクトリを基準にして、ディレクトリの経路を短縮する事ができます。
このように、ディレクトリまでの経路の表しかたは、ほかのコマンドでも、使う事ができます。

例えば

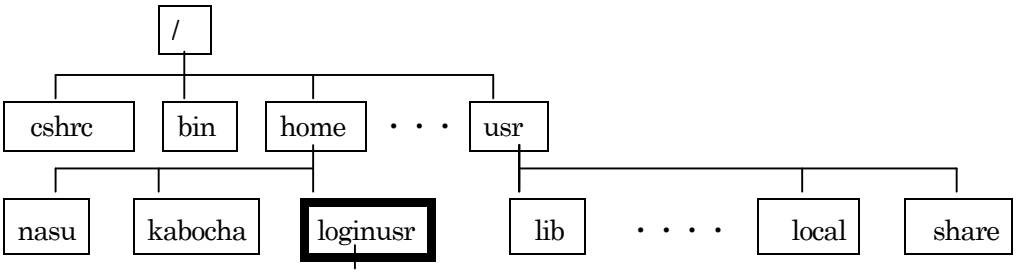
cd ..　，cd

ホームディレクトリ　/home/loginusr から　/　（ルートディレクトリ）まで、カレントディレクトリを、変更するには、**cd ../../** で　/ に　カレントディレクトリを変更できます。



最後にホームディレクトリで今まで作ったディレクトリを全て削除します。
pwd でカレントディレクトリを確認。
ls で　/home/loginusr のディレクトリのなかのディレクトリを確認。
rmdir dirctory1　とコマンドを打って　/home/loginusr/dirctory1 を削除します。
ls で　/home/loginusr のディレクトリの中に　ディレクトリ　dirctory3 が残っている事を確認。
次に、/home/loginusr/dirctory3/dirctory3-3 を削除する。
ls で　ディレクトリ/home/loginusr/dirctory3 の中身を確認する。
dirctory3-1　　dirctory3-2　　dirctory3-3　がある事を確認する。
rmdir dirctory3/dirctory3-3　で　/home/loginusr/dirctory3/dirctory3-3 を削除する
ls で　ディレクトリ/home/loginusr/dirctory3 の中身を確認する。
dirctory3-1　　dirctory3-2 がある事を確認。
最後に　/home/loginusr/dirctory3 とその下の階層のディレクトリ　/home/loginusr/dirctory3/dirctory3-2 と
　　/home/loginusr/dirctory3/dirctory3-1 をまとめて削除する。
rmdir dirctory3 を実行します。しかし　/home/loginusr/dirctory3 の下には、ディレクトリがある為に、削除する事ができません。そんな時は、
rm –r　または、　rm –R　でまとめて　削除することができます。

rm –R　 dirctory3 で　まとめて削除します。
ls　で　ディレクトリ　 dirctory3 の中に何もないのを確認して、終わりです。




```
% pwd
 /home/loginusr
% ls
   dirctory1              dirctory3
% rmdir dirctory1
%ls
% dirctory3
%ls dirctory3
   dirctory3-1          dirctory3-2          dirctory3-3
% rmdir dirctory3/dirctory3-3
%ls dirctory3
   dirctory3-1       dirctory3-2
% rmdir dirctory3
rmdir: dirctory3:/Directory not empty
% rm –R dirctory
% ls
%
```

rmdir [オプション]　ディレクトリ名

空のディレクトリ fdir1 を削除
%rm fdir1

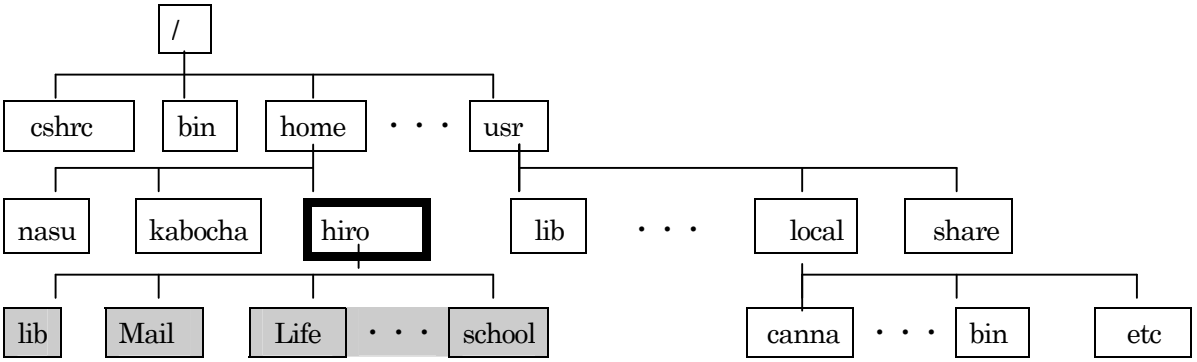
ディレクトリ　fdir2 の中身を丸ごと削除
% rm –r fdir2

ls

ディレクトリにあるファイルを表示してみましょう

ファイルを表示するコマンドは　ls　です。　ｌｓ　は　list の略です。
ｌｓ コマンドを実行してみます。

```
% ls
lib       Mail       Life     ・・省略・・・     school
%
```

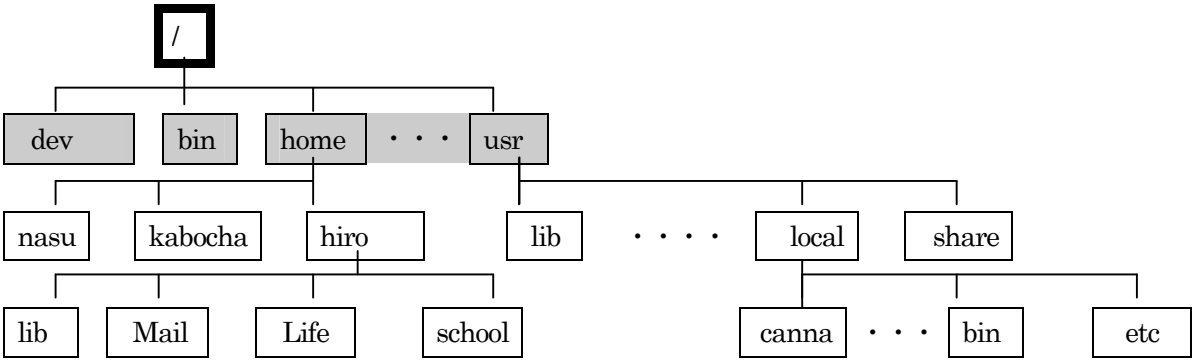


（　・・・　は　途中ファイル省略　）

これは、hiro　が、ログインした時の　ディレクトリを表しています。
ログインした時ユーザは、自分のホームディレクトリと呼ばれるディレクトリにいます。
自分のディレクトリの中のファイルを自由に編集したり、削除したり、ファイルを入れたり、出したりする事ができます。

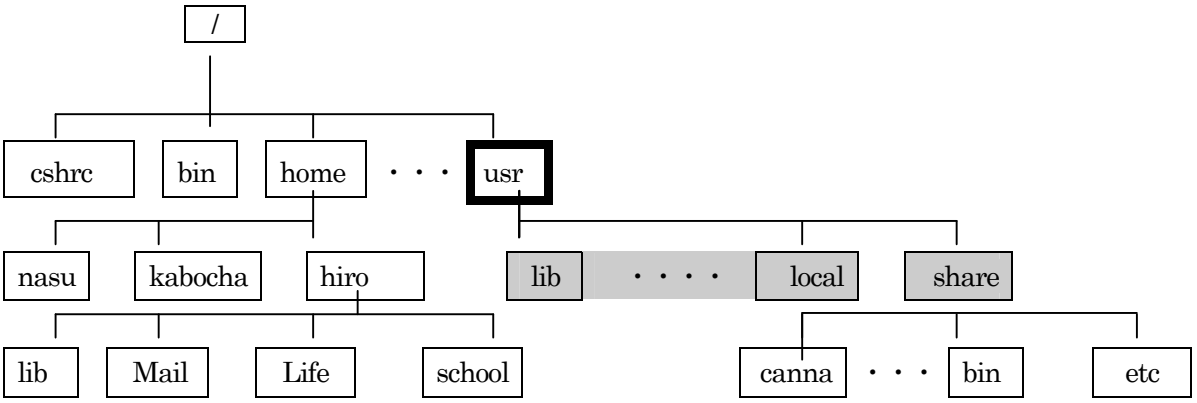
つぎは、根っこの部分の　/　にあるファイルを表示してみましょう。

```
% ls /
COPYRIGHT        boot              dist              lib          ・・・
 usr             cdrom             entropy           libexec      ・・・
dev              compat            etc               media        ・・・
bin              home              mnt          ・・・
%
```



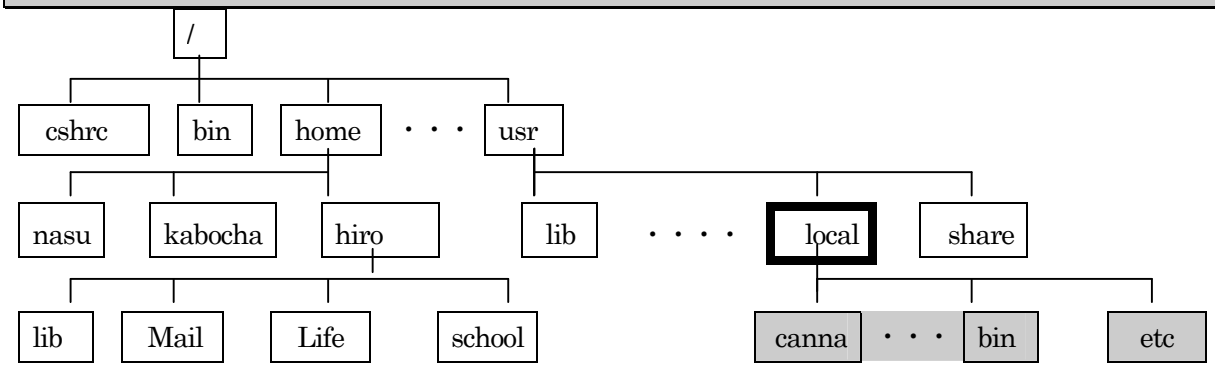
次は / の中の usr というディレクトリの中を表示します。

```
% ls /usr/
  libate          X11R6          libexec        src
 bin              local          compat         obj
games             ports          home           sbin
 include          share          lib            %
```



次は / の中の usr のなかの local というディレクトリの中を表示します

```
% ls /usr/local
 bin          man          lib            libadata
 etc          share        libxeec        include
  info        canna        sbin           build
%
```



次は　–F　オプションをつけて　ls を実行してみます。
–F　オプションにより分類記号がファイル名に付け加えらる記号です。

/	ディレクトリ
@	シンボリックリンク
*	実行可能ファイル

```
% ls –F /usr/
    libate/          X11R6@          libexec/          src/
    bin/             local/          compat/           obj/
     games/            ports/           home/             sbin/
    include/         share/          lib/
%
```

次は　–a オプションをつけて実行してみます。

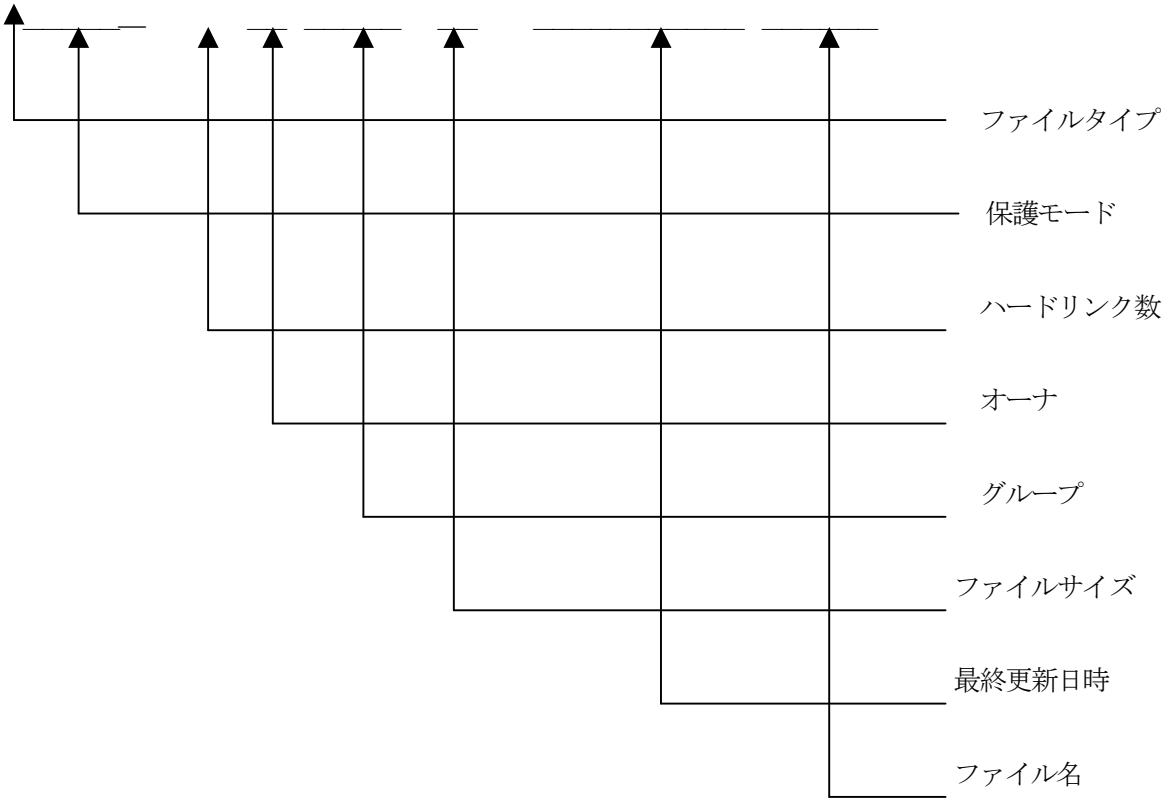
```
% ls –aF /usr/
   ./          libate/          X11R6@          libexec/          src/
   ../            bin/             local/             compat/          obj/
   .snap/       games/           ports/           home/             sbin/
   include/       share/           lib/
%
```

-a　オプションを付けることで、” . ”(ドット)で始まる。ディレクトリやファイル(通称ドットファイル)を表示する事ができます。オプションなしでは、表示しません。

” . ” はカレントディレクトリをあらわし、” .. ”は、親ディレクトリを表しています。

次は、ファイルやディレクトリの詳しい表示をするために –l オプションをつけて実行です。

```
%ls –al /usr
 total 56
drwxr-xr-x   17 root  wheel   512     Jun   21   17:11   .
drwxr-xr-x   20 root  wheel   512     Aug   15   09:20   ..
drwxr-xr-x    2 root  operater 10     Jun   21   18:31   .snap
lrwxr-xr-x    1 root  wheel    10     Jun   21   21:17   X11R6      ->   /usr/local
```



ファイルタイプには、 “d” ,(ディレクトリ), “l”(シンボリックリンク), “-“(普通のファイル)がある。

ｌｓ　[オプション]　[ディレクトリ　又は　ファイル名]

-a　ファイル／ディレクトリ　プラス(+)　ドットファイルを表示
-F　分類記号　　/ディレクトリ、@シンボリックリンク、*実行ファイルを加えて表示
-l　更新日時やファイルやディレクトリのオーナ、保護モードを表示
-t　ファイルやディレクトリをアルファベット順ではなく、更新日時に従って順に表示

cat

cat コマンドは、ファイルの中身をみるコマンドですが、下記のようにファイルを作ってみましょう。

ファイル名は　　aisatu1
　中身の文は　　ohayougozaimasu
　　aisatu1 は（あいさつ 1）のことで　ohayougozaimasu は(おはようございます)のことです。

まず、　cat > aisatu1　と文字を打ってみましょう。

```
% cat > aisatu1
```

aisatu1 の文字をうちこみましたら、Enter キーを押します。

次に、中身の文の　ohayougozaimasu の文字を入力して、Enter キーを押します。

```
% cat > aisatu1
  ohayougozaimasu
  ■
```

の画面になりましたら

Ctrl + D　（コントロールキーを押しながら d を押す）を行います。

```
%cat > aisatu1
  ohayougozaimasu
  %D
```

上記の様に表示されます。

次に、aisatu1 と言うファイルがあるかを確認します。

```
% ls
aisatu1
```

ファイル名　aisatu1　を確認できましたら、　ファイルの中身を見ましょう。

cat aisatu1 と入力して Enter キーを押します。

```
% cat aisatu1
    ohayougozaimasu
%
```

ファイル名　aisatu1　の中身　ohayougozaimasu　が見れましたら、成功です。

ｃat コマンドは、テキストファイルの中身を見る事ができるのですが、
バイナリファイルは、テキストファイルではないので、中身を見ようとすると、訳が解らない文字列が表示されてしまいます。

バイナリファイル：（binary file）専用のアプリケーションソフトで扱うことを前提として、文字コードの範囲などを考慮せずに作成されたファイルのこと。画像や動画、音声を記録したファイルや、実行可能形式のプログラムを収めたファイルなど、文字のみで構成されるテキストファイル以外はすべてこれに含まれる

次は

ファイル名は　　aisatu2
　中身の文は　　yorosikuonegaisimasu
　　aisatu2 は（あいさつ 2）のことで　yorosikuonegaisimasu は(よろしくおねがいします)のことです。

まず、　cat > aisatu2　と文字を打ってみましょう。

```
% cat > aisatu2
```

aisatu2 の文字をうちこみましたら、Enter キーを押します。

次に、中身の文の　yorosikuonegaisimasu の文字を入力して、Enter キーを押します。

```
% cat > aisatu2
  yorosikuonegaisimasu
  ■
```

の画面になりましたら

Ctrl + D　（コントロールキーを押しながら d を押す）を行います。

```
% cat > aisatu2
  yorosikuonegaisimasu
  %D
```

上記の様に表示されます。

次に、aisatu1 と言うファイルがあるかを確認します。

```
% ls
aisatu1      aisatu2
```

ファイル名　aisatu2　を確認できましたら、　ファイルの中身を見ましょう。

cat aisatu2 と入力して Enter キーを押します。

```
% cat aisatu2
    yorosikuonegaisimasu
%
```

ファイル名　aisatu2　の中身　yorosikuonegaisimasu　が見れましたら、成功です

次に、2 つのファイル　aisatu1　と　aisatu2 のファイルを同時に、　見てみましょう。

```
% cat aisatu1 aisatu2
ohayougozaimasu
yorosikuonegaisimasu
%
```

cat の、コマンド名は　（ concatenate 連結する　）の略です。

ｃat コマンドは、１画面しか表示できません。そんな時は、cat　の代わりに
more コマンドをつかいます。

more /COPYRIGHT と入力してみましょう。

```
%more　/COPYRIGHT
```

すると、画面いっぱいに英語で、一画面では表示しきれずにいます。
Enter キーを押すと１行ずつ下に移動します。矢印キー　↓　↑　でも　移動できます。

cat　[オプション] [ファイル１　ファイル２・・・ファイル n]

オプション
-n　表示する内容に行番号をつける。
-b　表示する内容に行番号をつけるが、空白の行は、カウントしない。

例
ファイル　aisatu1 の内容を表示
% cat aisatu1

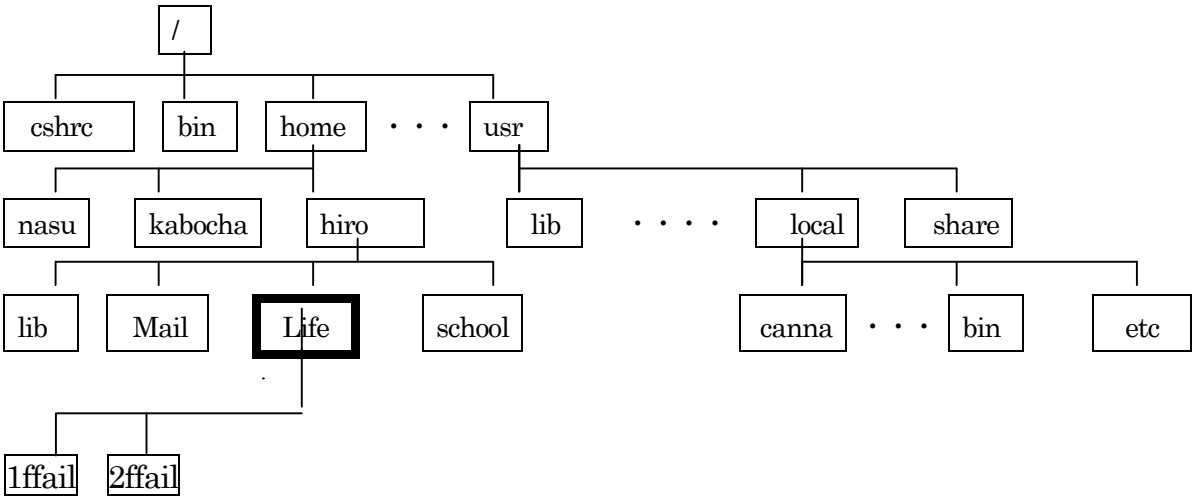
ファイル　aisatu1 aisatu2 aisatu3　を連結して表示
% cat aisatu1 aisatu2 aisatu3

標準入力した内容を表示
% cat
abc　　（と入力して　Enter キーを押すと）
abc　　（と表示される。）
orange　（と入力して　Enter キーを押すと）
orange　（と表示される。）

Ctrl + D(Ctrl キーを押しながらｄを押す。）をすると、終了です。

リダイレクトで、ファイル aisatu4 をつくる。
% cat　> aisatu4　　(Enter キーを押す。)
hajimemasite　　(Enter キーを押す。)
Ctrl + D(Ctrl キーを押しながらｄを押す。）をすると、ファイル　aisatu4 をつくります。
↓
% cat aisatu4
hajimemasite
%

copy



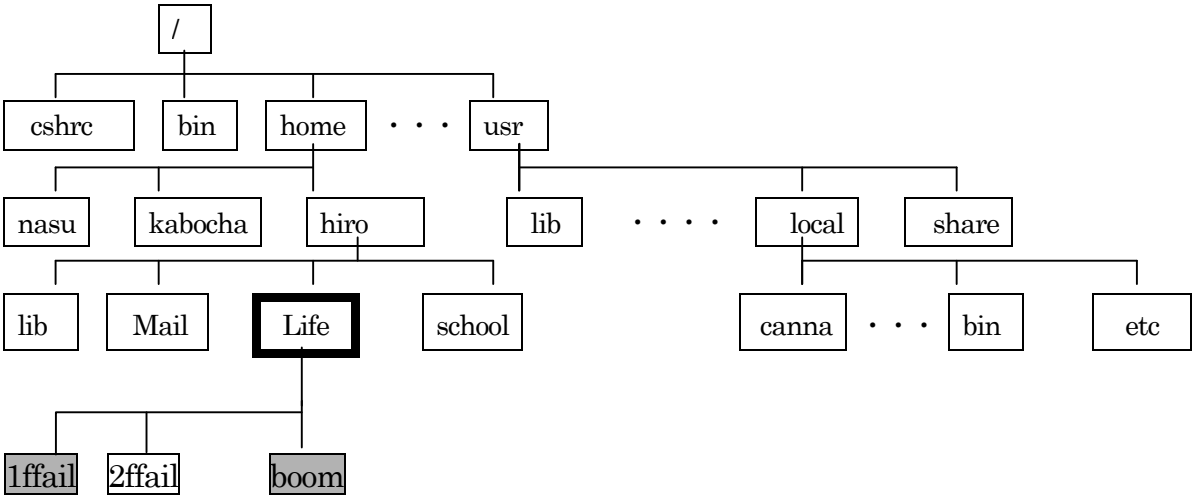
ファイルをコピーします。　コピーのコマンド書式は

cp コピー元ファイル名　　新コピーファイル名

ファイル　1ffail をコピーして　boom と言うファイルをつくりましょう。

```
% pwd
/home/hiro/Life
% ls
1ffail      2ffail
% cp 1ffail boom
% ls
boom      1fail     2fail
%
```

まず、現在のカレントディレクトリを確認します。
次に　ls　コマンドで、現在あるファイルを確認します。
ファイル名　ffai1 を　boom　と言うファイル名でコピーする。
ｌｓ コマンドで　コピーファイル名　boom　が有るかを　確認する。




```
% cat 1ffail
  good
% cat 2ffail
 mornig
% cat boom
  good
%
```

ファイル 1ffail の内容を確認する。
ファイル 2ffail の内容を確認する。
ファイル boom の内容がファイル 1ffail の内容と同じ事を確認する。

ｃｐ　コマンドで、危険なことは、誤ってファイルを上書きしてしまう事です。

続いて
誤ってファイルを上書きしてしまうケースを試してみましょう。

ファイル boom の 中身は、goo　でした。
しかし、boom が作成されたあとに、時が経って、ls コマンドを使用しないで、boom　の存在をわすれてしまい。
ファイル 2ffail　をコピー元にして　boom を作成するために
cp　2ffail boom　　とコマンドを打ってしまった時に　boom の内容が　どうなるか？　を実験してみてください。

```
% ls
   1ffail        2ffail          boom
% cat boom
   good
% cp    2ffail boom
% ls
   1ffail        2ffail          boom
% cat boom
   mornig
%
```

ファイル名　boom　の存在を知らずに、cp コマンドをしようして　ファイル名 boom を作成してしまうと、上書きされてしまう事がお解りになると思います。
そのように、大変危険なコマンドなので、気をつけて使わないといけません。

cp コマンドには、　-i オプション があるので、危険を避ける工夫があります。
cp -i コマンドを実行してみます。

```
% pwd
 /home/hiro/Life
%   ls
   1ffail       2ffail     boom
% cp    -i 1ffail boom
   overwrite boom?  (y/n [n]) n
   not overwritten
% cp -i 1ffail boom
   overwrite boom?  (y/n [n]) y
%
```

pwd カレントディレクトリを調べます。
ｌｓで　ファイルの確認をします。
cp -i 1ffail boom で ファイル 1ffail をコピー元にして　ファイル boom をコピーしようと実行します。
overwrite boom? (y/n [n])　の表示がして、boom を上書きをするのか　y か n で答えろと表示されます。
overwrite boom? (y/n [n])**n**　と　n を入力して Enter キーを押すと
not overwritten　と　上書きできなかったと表示されます。
もう一度　cp　-i　1ffail boom を実行します。
おなじように overwrite boom? (y/n [n])　の表示をされます。
overwrite boom? (y/n [n])**y**　と　y を入力して Enter キーを押すと
上書きされます。
パスを使って　cp コマンドを使用する事ができます。

ｃｐ　[オプション]　コピー元ファイルn[ディレクトリ]　コピー先ディレクトリ

書式

1番目	cp [-i -f -R -P　]	source_file	target_file
2番目	cp [　-R -P　]	source_file……	target_directory

1番目の書式の場合、cpは　source_fileの内容を　target_fileにコピーします。
2番目の書式の場合、cpは　source_fileの各ファイルがtarget_directoryの中へコピーされる。
このとき名前は変更されない。

-i　コピー先に同じ名前のファイルが、在る時には、上書きするのか、しないのかを　確認をユーザにする。
確認をしてから、上書きをするか、しないかを　実行します。

-f　コピー先に同じ名前のファイルが、在る時、そのファイルのパーミッションに関わらずに確認をもとめないで、すでにある、コピー先の同じ名前のファイルを消去して、新しくファイルをつくる。

-R　source_fileとしてディレクトリが指定された時、そのディレクトリと、そのディレクトリ以下の部分木を構成する全てのファイルをコピーする。

-P　ファイルの変更時刻、アクセス時刻、フラグ、モード、ユーザID、グループIDなどを、パーミッションが許す範囲内で、可能な限り保存してコピーする。

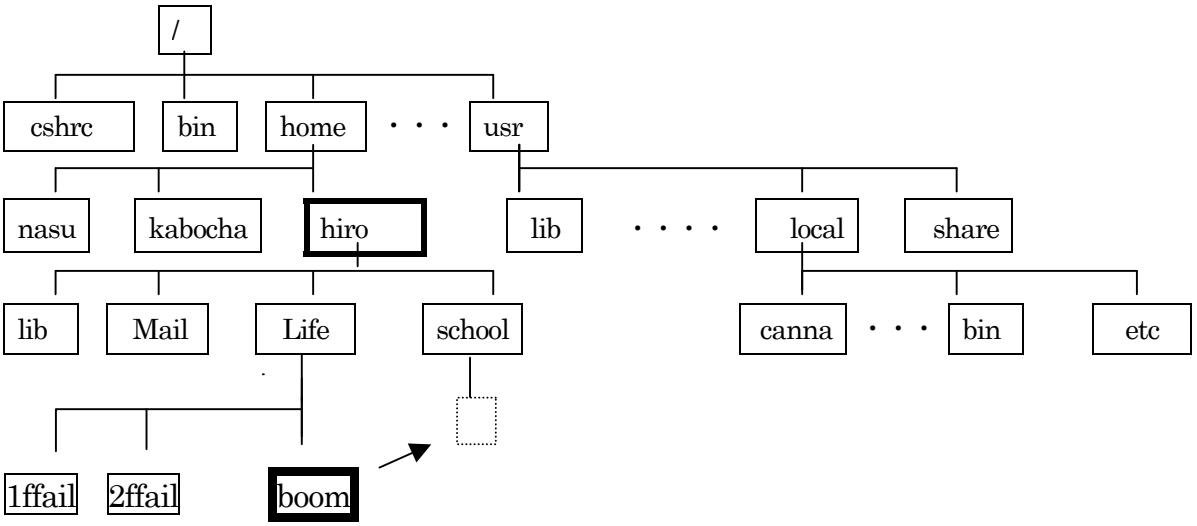
パーミッション：(permission)
コンピュータのハードディスクなどに保存されているファイルやディレクトリに対するユーザのアクセス権のこと。一般に、UNIXシステムにおけるアクセス権を指す言葉として用いられる。
UNIX システムにおけるパーミッションは、ファイルディレクトリの所有者である「Owner」、同じマシンを利用できるユーザ全体を意味する「Group」、その他全ての「Other」に対して、それぞれ「読み込み」「書き込み」「実行」の権限を設定できる。

m v

mv コマンドを使って、ファイルを移動してみましょう。

mv　(move) 移動
mv　移動元ファイル名 n　　移動先ディレクトリ

カレントディレクトリ hiro の中のディレクトリ Life のなかのファイル boom を
カレントディレクトリ hiro の中のディレクトリ school のなかのファイル boom として移動します。

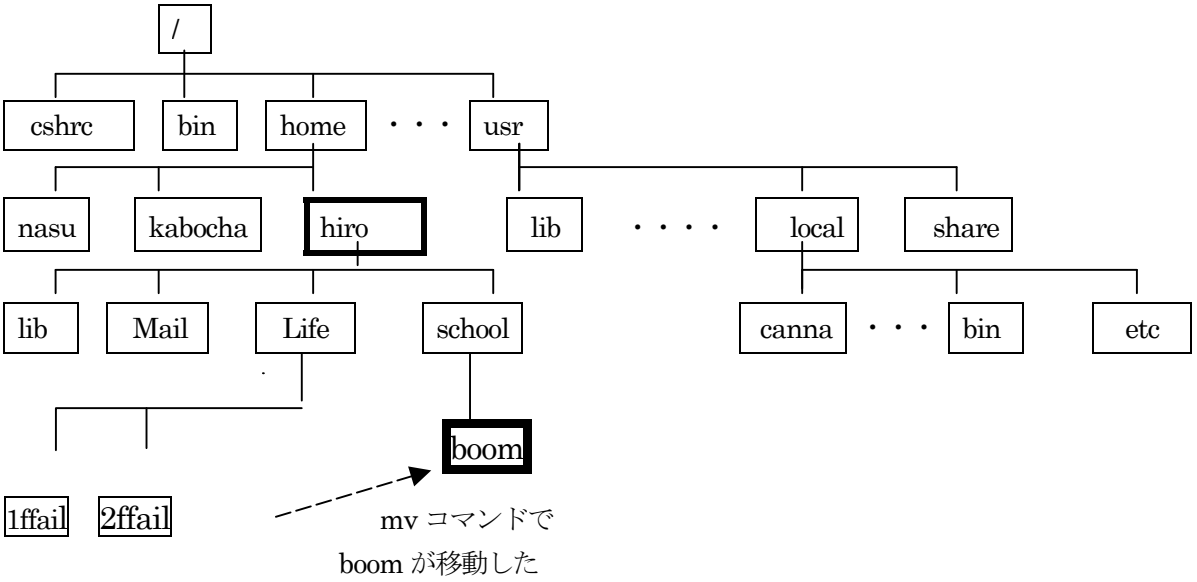


ファイル boom を school
のツリー構造の部分木に
するために移動

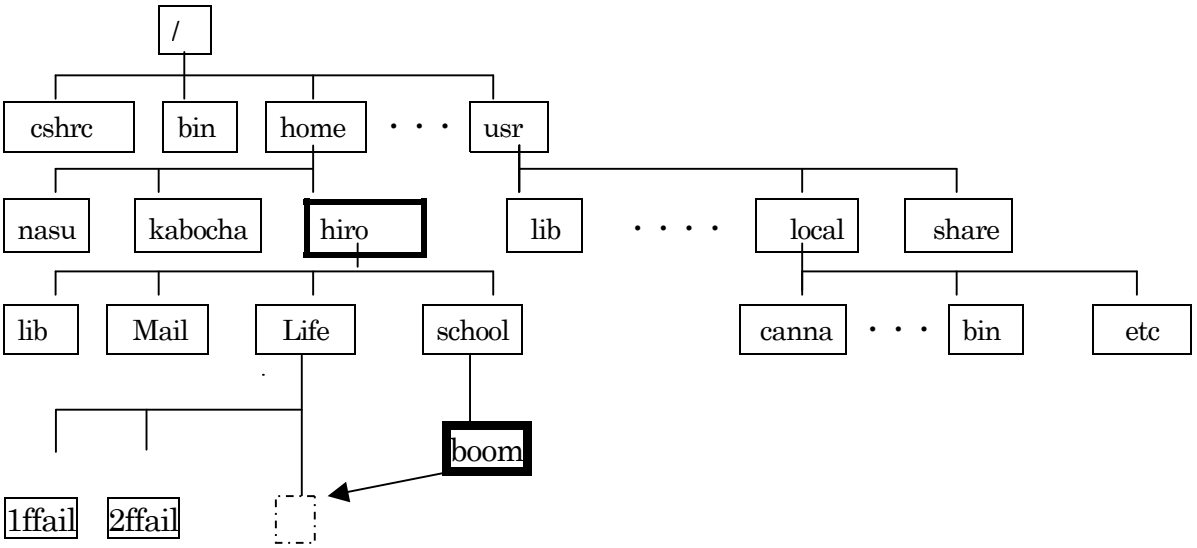
```
%　pwd
/home/hiro/
%　ls Life
　1ffail　2ffail　boom
%　mv Life/boom　school
% ls school
　boom
% ls Life
　1ffail　　2ffail
%
```

まず pwd コマンドで　/home/hiro と　カレントディレクトリの確認をします。
ｌｓ　Life で　/home/hiro/Life のなかの　1ffail,　2ffail,　boom　ファイルを確認。
mv　Life/boom school　で　mv コマンドを使い　ファイル/home/hiro/Life/boom を
ディレクトリ/home/hiro/school のツリー構造の部分木にする。
ｌｓ school で　ディレクトリ school に　ファイル boom　が有るのを、確認。
ｌｓ　Life　で　ファイル boom　が無い事を確認。

下記の図ｙのように、ファイル boom がディレクトリ school のツリー構造の部分木ファイル boom なります。



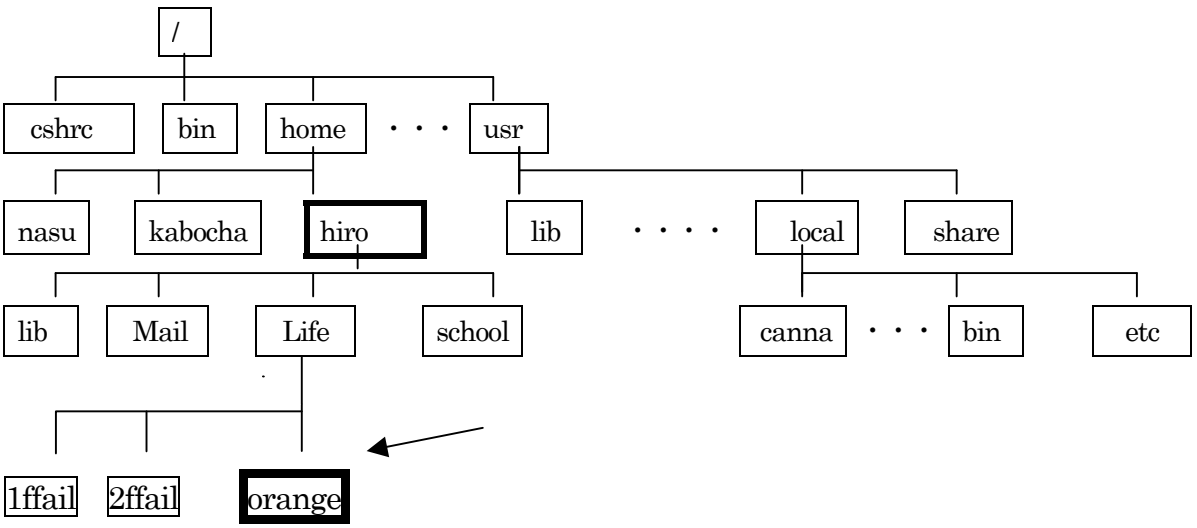
こんどは、今とは、逆にファイル/home/hiro/school/boom を ファイルの名前を変えて
ディレクトリ/home/home/Life の部分木ファイル名 orange にして移動します。



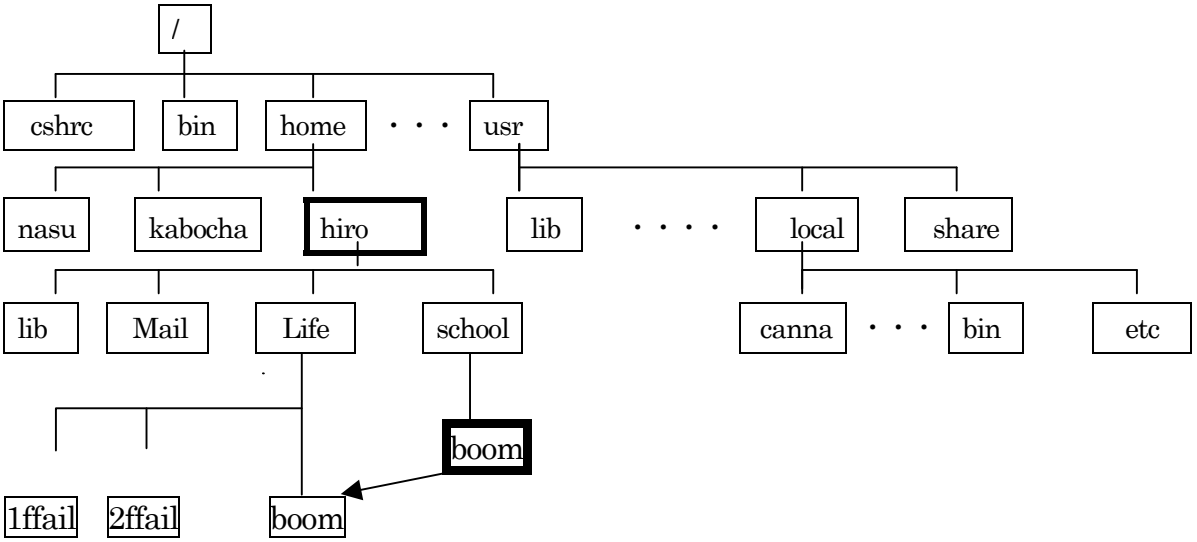
```
%   pwd
/home/hiro/
%   ls Life
    1ffail    2ffail
%   mv school/boom   Life/orange
% ls school
%
% ls Life
   1ffail    2ffail   orange
%
```

まず pwd コマンドで　　/home/hiro と　カレントディレクトリの確認をします。
ｌｓ　Life で　/home/hiro/Life のなかの　1ffail,　2ffail　ファイルを確認。
mv　school/boom Life　で　mv コマンドを使い　　ファイル/home/hiro/school/boom を
ディレクトリ/home/hiro/Life のツリー構造の部分木にして、かつファイル名を orange にする。
ｌｓ school で　ディレクトリ school に　ファイル boom　が無いのを、確認。
ｌｓ　Life　で　ファイ 1ffail,　2ffail,　orange ファイルを確認

下記の図のようにファイル boom が　ディレクトリ Life のツリー構造の部分木になり、かつ、ファイル名が orange
になります。



mv コマンドにも　　cp コマンド同様に　-i オプションがあります。 同じファイル名が在るディレクトリに　ファイルを移動する時に、上書きの確認注意をうながします。



上記の図のように、ディレクトリ/home/hiro/Life にファイル名 boom があるのに
ディレクトリ/home/hiro/school のファイル名 boom を
ディレクトリ/home/hiro/Life に移動しようとすると、ディレクトリ/home/hiro/Life にファイル名 boom が上書きされてしまいます。そのような時に

mv -i /home/hiro/Life/boom　/home/hiro/Life と　オプション　-i　をつける。

```
%ls Life
   1ffail       2ffail       boom
% mv –i   /home/hiro/school/boom    /home/hiro/Life
   overwrite boom?  ( y/n [n])n
   not overwritten
%
```

書式

ｍｖ　[オプション] 移動元ファイルｎまたは[ディレクトリ]　　移動先ディレクトリ

オプション

-i　同名のファイルが　移動先にある時には、上書きするかどうかを確認注意をうながします。

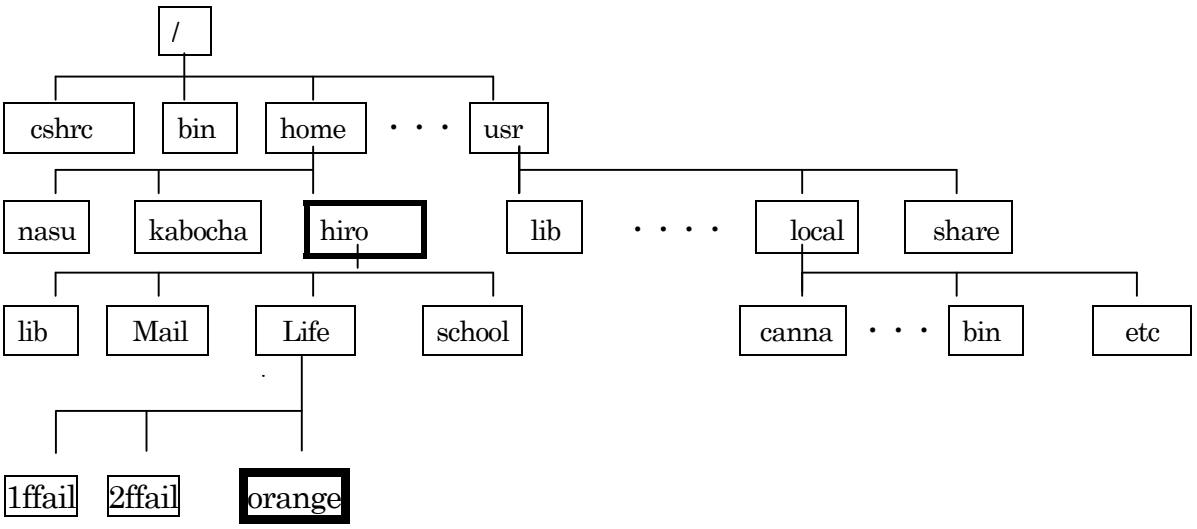
-f　-i オプションとは反対に確認をしないで、全ての上書きをおこないます。

rm

rm コマンドはファイルを削除します。

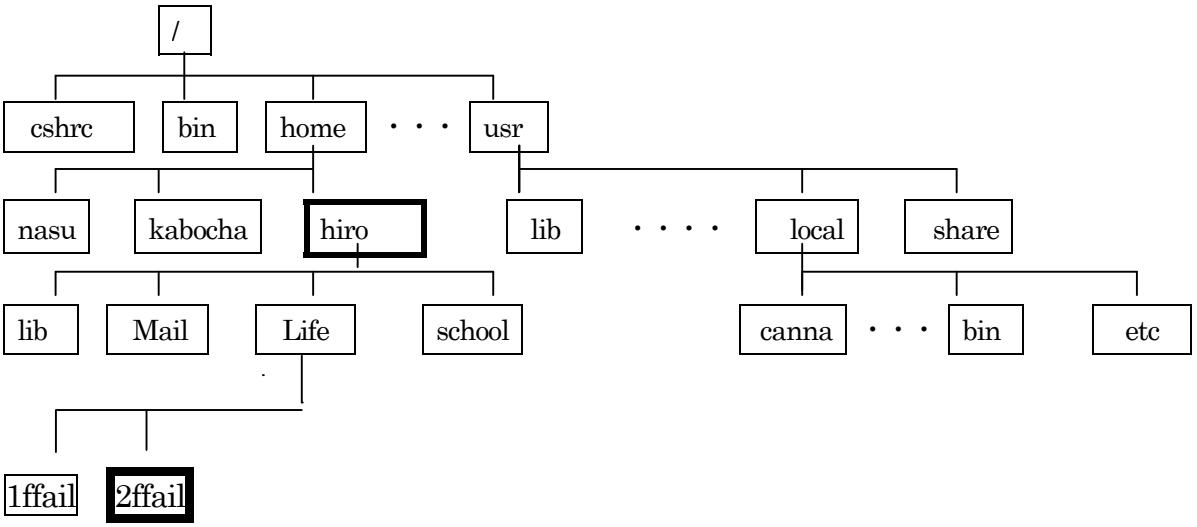
rm ファイル名
と入力したら　ファイルが削除されます。

まず　下記の図の
ファイル/usr/home/hiro/Life/orange を削除してみましょう。




```
% pwd
  /home/hiro
% ls Life
  1ffail     2ffail     orange
% rm   orange
% ls
    1ffail       2ffail
%
```

pwd で　カレントディレクトリを確認
ls　Life で　/home/hiro/Life のなかの　1ffail　　2ffail　　orange　　ファイルを確認
rm orange で　ファイル orange を削除
ls で orange が削除されているのを確認する。



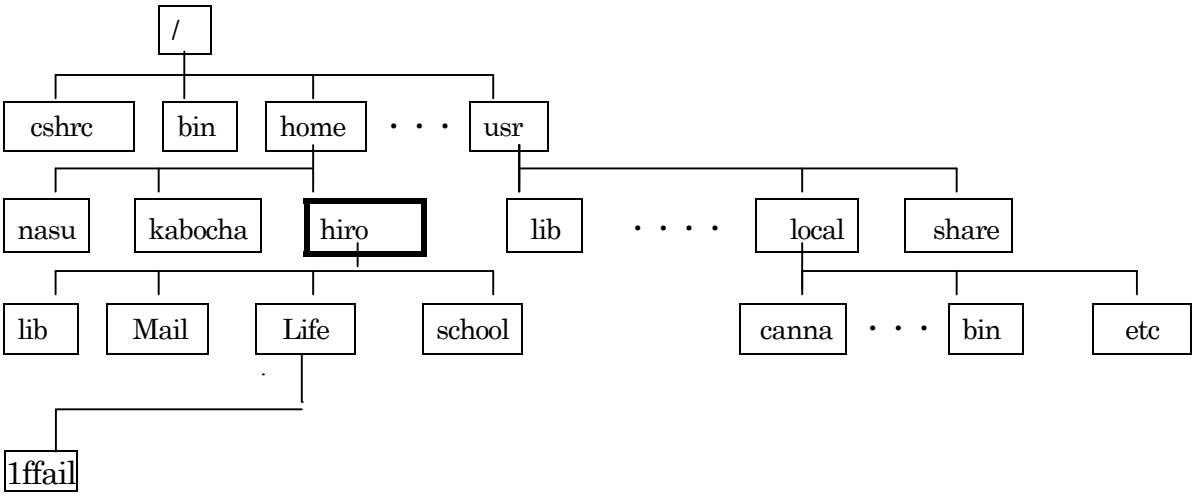
rm コマンドにも　-i オプションがついています。

ファイル/home/hiro/Life/2ffail を　-i オプションを使って削除します。

```
%pwd
 /home/hiro
% ls Life
    1ffail    2ffail
% rm -i   2ffail
 remove 2ffail? n
% ls Life
    1ffail   2ffail
% rm -i   2ffail
  remove 2ffail? y
% ls Life
  1ffail
%
```

pwd でカレントディレクトリを確認
ls Life で　/home/hiro/Life のなかのファイルを確認
rm　–i　　オプションをつかってファイル 2ffail を削除
remove 2ffail?　と削除確認を出してもらい、n を押す。
ls Life で、/home/hiro/Life の中のファイルが　1ffail ,　2ffail であり、2ffail が削除されてない事を確認。
remove 2ffail?　と削除確認を出してもらい、y を押す。
ls Life で、/home/hiro/Life の中のファイルが　1ffail　だけで、2ffail が無い事を確認

下記の図のように、ファイル 2ffail が削除されました。



書式

rm　[オプション]　　ファイルｎ

rm　[-f　| -i　] [-d -r -R]　　file….

-f　:-i オプションとは反対にユーザに削除の確認をしないで削除する。

-i :削除するのか、削除しないのかを　ユーザに確認をする。

-d：ディレクトリも、他のダイプのファイルと同じように削除する。-d オプション無しにファイルとしてディレクトリを指定するとエラーになる。

-r　,　–R：引数　file として指定したディレクトリを rm コマンド自身がディレクトリを削除する時に処理が終了するまでに、何度も呼び出して繰り返し処理をする。-r　,　–R オプションは、自動的に、-d オプションが指定される。-i オプションが指定された時、最初にファイルの削除をする、しないの確認をもとめられる。
さらに、ディレクトリの奥のディレクトリについて、ディレクトリの中身の削除の前に確認がもとめられる。確認に対して削除すると、答えなかったときには、それ以下のディレクトリの削除はおこなわれない。

使用例

ファイルをまとめて削除する。

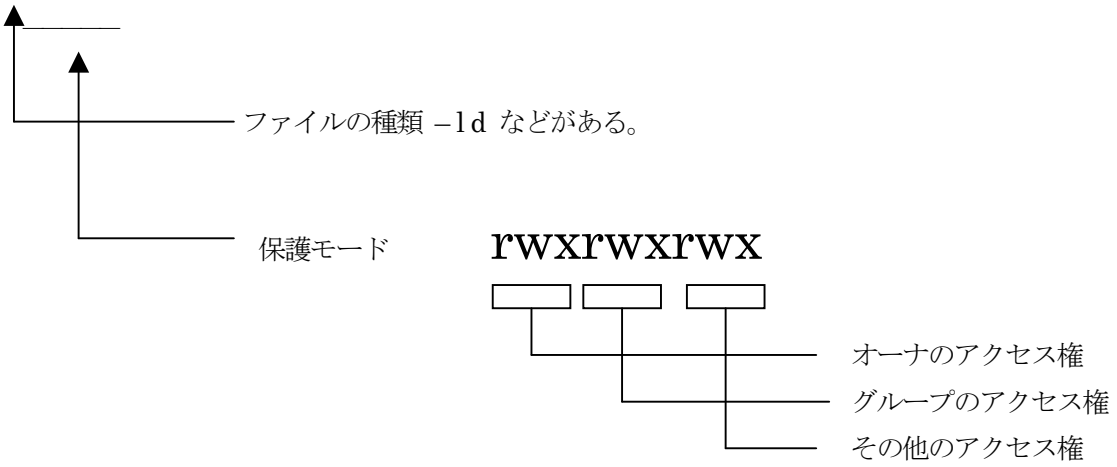
ffail , ffail2, ffail3, ffail4 をまとめて削除

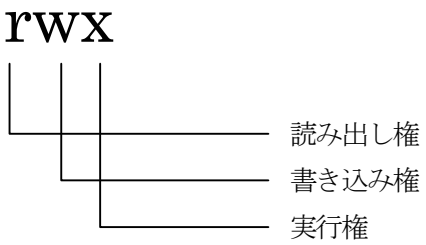
rm ffail　ffail2　ffail3　ffail4

ファイルのアクセス権

FreeBSD は、マルチユーザシステムなので、複数のユーザが一つのコンピュータシステムを利用します。個人使用のファイルに、他のユーザが書き込みをしたり、勝手にコピーをされたり、削除されたりしたら、大変困ります。そのような事が起こらないように、ファイルのアクセス権を、設ける事ができます。

```
%ls –al /usr
 total 56
drwxr-xr-x   17 root  wheel   512     Jun   21   17:11   .
drwxr-xr-x   20 root  wheel   512     Aug   15   09:20   ..
drwxr-xr-x    2 root  operater 10     Jun   21   18:31   .snap
lrwxr-xr-x    1 root  wheel    10     Jun   21   21:17   X11R6        ->   /usr/local
```





ファイルの種類は、－はファイル，ｌは、リンク　ｄは、ディレクトリ　ｓはソケット

保護モードは、オーナ、　グループ　、その他　の順番に
ｒ：読み出し　　ｗ：書き出し　　ｘ：　実行の許可条件が、表示されます。

例１　**－rwxr―r―**

この場合ファイルの種類はファイル、オーナは書き込み、読み込み、実行の許可があり、
グループは、読み込み許可あり、その他は、読み込み許可あり、と読みます。

例2　―rw―rw－rw－

この場合ファイルの種類ファイル、オーナは書き込み、読み込み許可あり
グループは、読み込み、書き込み許可あり、その他は、読み込み、書き込み許可あり、読みます。

アクセス権を変更するコマンドは
chmod (change mode)

chmod　ファイルモード　ファイル名

誰に　　：(オーナ、グループ、その他,　全て　）に、（ u , g , o, a）
加減　　：(加える、　減らす , 新設定)　　（　+, -, = ）
どのアクセス権　：(読み出し、書き込み、実行）を　（r, w, x)

まず、aisatu1 というファイルを作ります。

```
% cat > aisatu1
  ohayougozaimasu
  ■
```

Ctrl + D　（コントロールキーを押しながら d を押す）を行います。

次に、ls　–l コマンドで、ファイルモードを見ます。
オーナの読み出し権を減らします。

```
% ls -l aisatu1
-rw-r--r--   1    hiro    user    16  Nov  21  08:23   aisatu1
% chmod u-r aisatu1
% ls -l aisatu1
--w-r--r--   1    hiro    user    16  Nov  21  08:25   aisatu1
% cat aisatu1
cat : aisatu1: Permission denied
%
```

まず、ls -l で　ファイル aisatu1 のアクセス権を表示します。
オーナ：書き込み、読み出し
グループ：読み出し
その他：読み出し
の確認をする。
つぎに、chmod コマンドで、オーナの読み込み権を減らす。
u-r　　オーナ、減らす（ー　マイナス）　読み込み権
ls -l で　ファイル aisatu1 のアクセス権を表示します
オーナ：書き込み、
グループ：読み出し
その他：読み出し
オーナの読み出しが減らされているのを、確認します。
cat コマンドで、ファイルをよみます。
否定されました許可を　と表示されました。

次は、グループとその他の読み出し権を減らします。

```
% ls -l aisatu1
--w-r--r--    1     hiro      user      16   Nov   21   08:30    aisatu1
% chmod    go-r aisatu1
% ls -l aisatu1
--w-------    1     hiro      user      16   Nov   21   08:35    aisatu1
%
```

ls –l コマンドでファイル aisatu1 のファイルアクセス権を確認
chmod コマンドで　ファイル aisatu1 のグループとその他の読み出し権を減らします。
ls –l コマンドでファイル aisatu1 のファイルアクセス権の確認をします。
グループとその他の読み出し権が減らされているのを、確認します。

次は、グループとその他の読み出し権を加えます。

```
% ls -l aisatu1
--w-------    1     hiro      user      16   Nov   21   08:40    aisatu1
% chmod    go+r aisatu1
% ls -l aisatu1
--w-r--r--    1     hiro      user      16   Nov   21   08:43    aisatu1
%
```

ls –l コマンドでファイル aisatu1 のファイルアクセス権を確認
chmod コマンドで　ファイル aisatu1 にのグループとその他の読み出し権を加えます。
ls –l コマンドでファイル aisatu1 のファイルアクセス権の確認をします。
グループとその他に読み出し権が加えられているのを、確認します。

８進数を使った保護モード

８進数	r　w　x
0	—　—　—
1	—　—　x
2	—　w　—
3	—　w　x
4	r　—　—
5	r　—　x
6	r　w　—
7	r　w　x

モードの設定は、　オーナ、グループ、その他　ごとに、
ｒｗｘ、ｒｗｘ、　ｒｗｘ、
８進数、８進数、　８進数
と、３つ並べます。

オーナが読み出しできるように、８進数をもちいて、保護モードを設定します。

```
% ls -l aisatu1
--w-------  1  hiro  user  16  Nov  21  08:35  aisatu1
% chmod 200 aisatu2
% ls –l
-rw-------  1  hiro  user  16  Nov  21  08:35  aisatu1
%
```

書式

chmod [オプション] モード設定　ファイル n[ディレクトリ n]

オプション

-R　　サブディレクトリ以下をまとめて、ファイルモードを変更。
-L　　-R オプションが指定されていれば、すべてのシンボッリクリンクをたどり、リンク先のオリジナルファイルの保護モードも、変更する。

保護モードの設定

誰に　　：(オーナ、グループ、その他,　全て　）に、(u , g , o, a)
加減　　：(加える、　減らす , 新設定)　　(　+, -, =)
どのアクセス権　：(読み出し、書き込み、実行）を　(r, w, x)

８進数をつかう

８進数	r	w	x
0	—	—	－
1	—	—	x
2	—	w	—
3	—	w	x
4	r	－	—
5	r	－	x
6	r	w	－
7	r	w	x

例
% chmod u+w　failmode
ファイル名　failmode の保護モードを、オーナに、書き込み許可を加える。

% chmod 624　failmode2
ファイル名　failmode2 の保護モードを、624 に変更する。
　オーナ：６（rw-）,グループ：２（-w-）,その他：４（r--）

　サブディレクトリ：(sub directory）あるディレクトリの中にあるディレクトリ

　シンボリックリンク：(symbolic link）ファイルやディレクトリに別の名前を与え、
　ユーザやアプリケーションがその名前を元ファイルと同様に扱えるようにする仕組み