

ファイルとディレクトリ

- ▶ 全てはファイルとして管理される
- ▶ ディレクトリ
- ▶ 絶対パスと相対パス

全てはファイルとして管理される

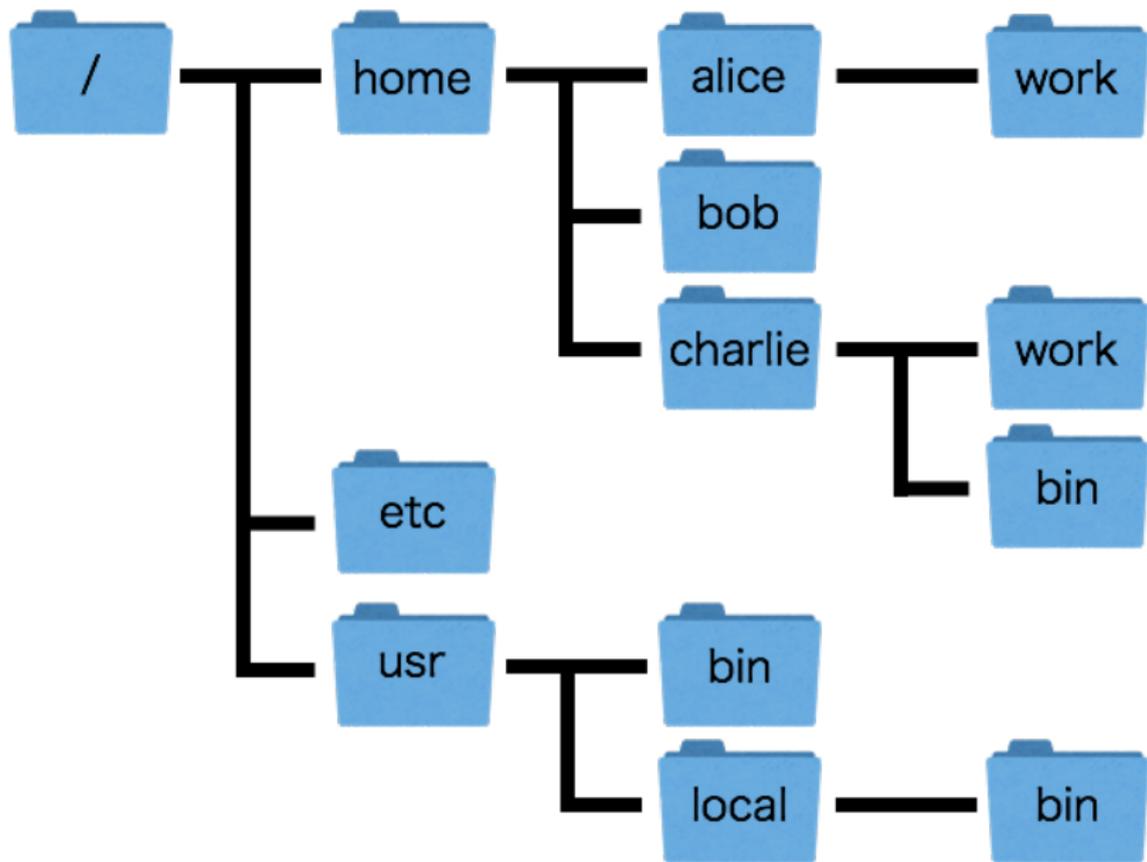
ファイル：データに名前を付けたもの

- ▶ 文書はファイル
- ▶ ディレクトリもファイル
- ▶ アプリケーションもファイル
- ▶ 周辺機器もファイル

ディレクトリ：ファイルを格納するファイル

- ▶ ファイルはディレクトリによって階層的に管理
- ▶ "/" ルートディレクトリを起点とした階層構造
 - 全てのファイルは / の下に存在する

ディレクトリの階層構造



ディレクトリ

住所みたいなものと思ったらよい

- ▶ `/usr/local/bin` ⇔ `/` の中の `usr` の中の `bin`
 - ▶ 先頭の `/` はルートディレクトリを表す
 - ▶ それ以外の `/` は切れ目を表す

- ▶ ルートディレクトリ : 全ての起点
- ▶ カレントディレクトリ : 現在地
- ▶ 親ディレクトリ : ひとつ上の階層
- ▶ ホームディレクトリ : 自分の家

カレントディレクトリ

現在地

- ▶ `pwd` : 現在地を表示するコマンド
- ▶ `cd` : 現在地を変更するコマンド

”.” はカレントディレクトリを表す

- ▶ カレントディレクトリが `/usr/local/bin` のとき、
.`.` は `/usr/local/bin` を指す

親ディレクトリ

ひとつ上の階層

- ▶ `/usr/local/bin` の親ディレクトリは `/usr/local`
- ▶ `/usr/local` の親ディレクトリは `/usr`
- ▶ `/usr` の親ディレクトリは `/`

".." はひとつ上のディレクトリを表す

- ▶ カレントディレクトリが `/usr/local/bin` のとき、
`..` は `/usr/local` を指す
- ▶ カレントディレクトリが `/usr/local` のとき、
`..` は `/usr` を指す
- ▶ カレントディレクトリが `/usr/local/bin` のとき、
`../..` は `/usr` を指す

ホームディレクトリ

ユーザ が自由に使用することのできる領域

- ▶ Linux はマルチユーザ
- ▶ 自分の領域，他人の領域，共有の領域
 - ▶ 利用に関する権限 ⇒ パーミッション
- ▶ 各ユーザ はそれぞれのホームディレクトリを持つ
- ▶ 通常， /home の下にあるユーザ名のディレクトリ
 - ▶ /home/george はユーザ george のホームディレクトリ

"~" はホームディレクトリを表す

- ▶ ユーザのホームディレクトリが /home/george のとき， ~ は /home/george を指す

ファイルを指定する方法

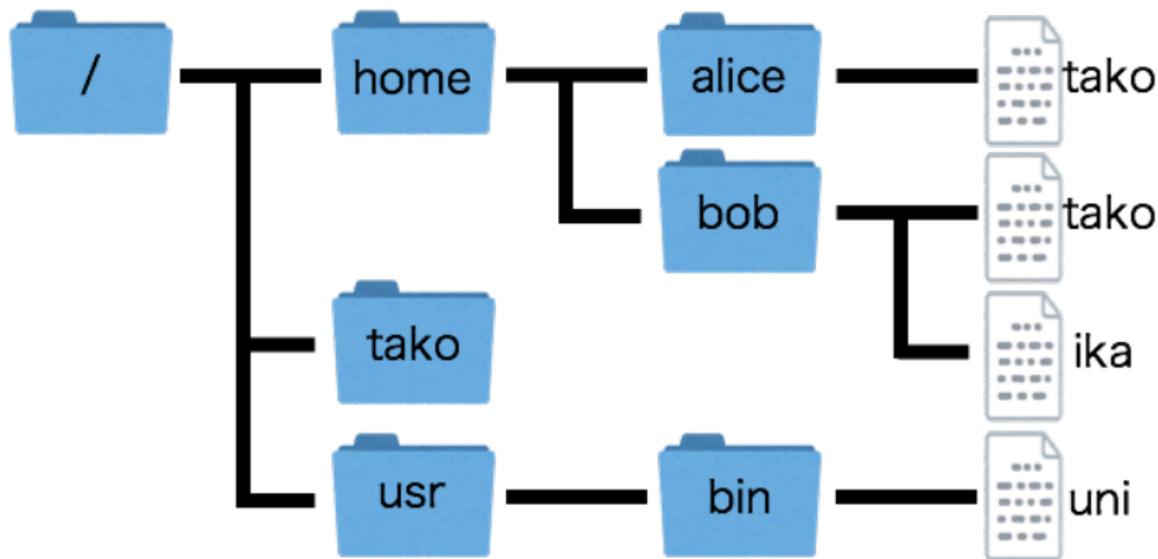
絶対パス

- ▶ ルートディレクトリを起点にして指定する

相対パス

- ▶ カレントディレクトリを起点にして指定する

絶対パスによるファイルの指定



▶ `/home/alice/tako`

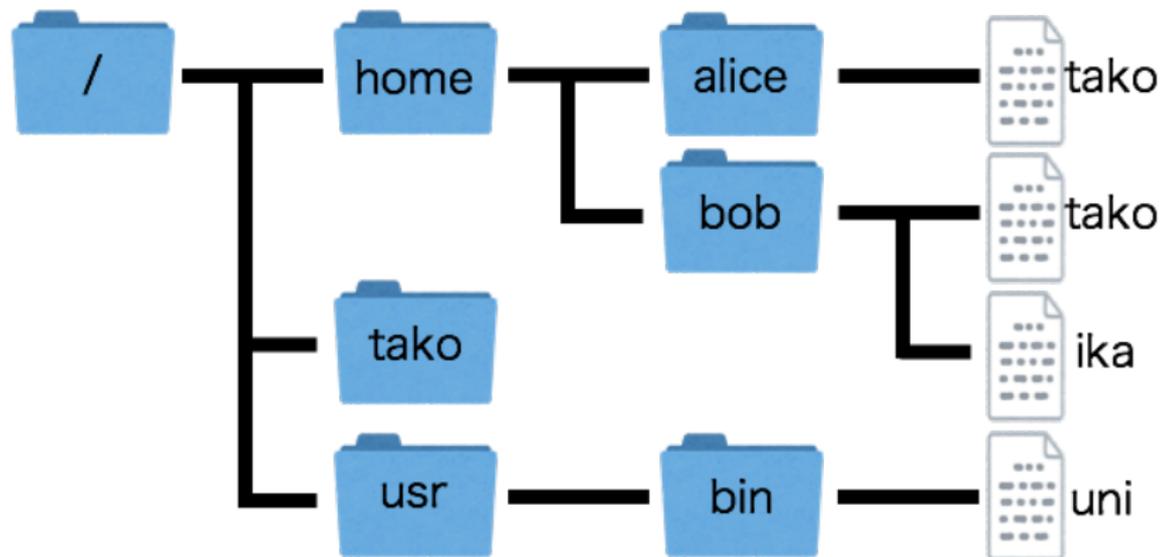
▶ `/home/bob/tako`

▶ `/tako`

▶ ファイル `ika`

▶ ファイル `uni`

相対パスによるファイルの指定



カレントディレクトリ `/home/alice` のとき

- ▶ `./tako`
- ▶ `../..tako`
- ▶ `../bob/tako`
- ▶ `../..usr/bin/uni`