

# 1 CUI を使った計算機利用とプログラミング

2

## 3 目的

4 CUI による計算機利用の概念を学ぶ

5 計算機のファイルシステムを学ぶ

6 プログラムの作成から実行までの過程を理解する

7

## 8 準備

9 端末を Linux で起動する。

10

## 11 Linux を利用した CUI 操作

12 教科書第 6・8 章も参照すること。

13

### 14 1. ログインと端末の起動

15 Linux の端末にログインする。

16 Linux では GUI が使えるシステムを X-Window システムと呼ぶ。

17 アプリケーション→アクセサリ→端末を開く

18 以下は端末のウィンドウで行う。

19

### 20 2. カレントディレクトリを表示

21 カレントディレクトリ(自分が今いるディレクトリ)を見る。

22 `pwd`

23 `pwd` はカレントディレクトリを表示させるコマンドで、おそらく present working

24 directory の略であろう。`/fs/home/nakakuki` のように表示される。Windows ではファ

25 イルをしまっておく入れ物はフォルダというが、Linux ではディレクトリと言う。

26 Windowsではフォルダの区切りは“¥”で表したが、Linuxでは“/ (スラッシュ)”である。

27 端末を開いたときに入っているディレクトリはホームディレクトリと呼ばれる。これ

28 が Windows のマイドキュメントに相当する。ファイルシステムはユーザー全員で共通

29 しているため、ホームディレクトリの中にのみファイルやディレクトリ(フォルダ)を作

30 ったり、消したりすることが出来る。

31

### 32 3. アプリケーションの実行とプロセス

33 ところで、Firefox は起動出来るだろうか？

34 `firefox`

35 と入力してみると、起動する。これは Linux システムがあらかじめ、アプリケーション  
36 を入れてある場所を知っているからである。`pwd`などのシステムのコマンドと同じ  
37 `/usr/bin`や`/bin`などの場所に入っている。コマンドは単機能のアプリケーションと思  
38 っても良いだろう。また、このとき、端末が入力できなくなっている。これは端末上で  
39 アプリケーションが走っているからである。このような実行をフォアグラウンド実行と  
40 いう。これだと不便なので、いったん `firefox` を終了する。`Ctrl(Control)`キーを押しな  
41 がら同時に `c` キーを押してみよう。

42 `Ctnl+c`

43 端末で `Ctrl+c` を押すと、フォアグラウンドで実行しているアプリケーションが強制終  
44 了される。この止め方はファイルを保存していない場合、作業が無駄になるから普段は  
45 使わない。アプリケーションが暴走したりして応答しなくなった時に使用する。もう一  
46 度、

47 `firefox &`

48 と入力する。今度は端末にコマンドが入力できる。このような実行をバックグラウンド  
49 実行という。走っているアプリケーションのことジョブあるいはプロセスといい、この  
50 場合バックグラウンドジョブあるいはバックグラウンドプロセスという。

51 `ps`

52 と入力すると、その端末上から走らせたプロセスを確認できる。`PID`の下にある番号は  
53 プロセス番号と呼ばれる。

54 `ps ux`

55 と入力すると、ユーザーが実行しているプロセスすべて、

56 `ps aux`

57 と入力すると、全てのユーザーが実行しているプロセスを確認できる。

58

#### 59 4. 環境設定の確認

60 `env`

61 と入力すると、ログインしているユーザーの環境設定が見られる。`enviroment`の略で  
62 ある。いろいろな設定があるが、`PATH`という変数に指定されているディレクトリが、  
63 コマンドやアプリケーションの入っている場所である。

64

#### 65 5. ディレクトリの作成

66 ディレクトリの中にあるファイルの名前をみる

67 ls

68 と入力する。list の略で、Windows の dir と同じ働きである。なお、Windows という  
69 ディレクトリがあるが、Windows のファイルはこちらに保存されている。

70 マウスで、デスクトップにある“userid(アカウント名)のホーム”のアイコンを開くと同  
71 じ名前のファイルが見える。表示を詳細表示にすると、ファイルの許可情報、容量や最  
72 終変更日時が見られる。これを端末で見ると、

73 ls -l

74 と入力する。こんな表示が出る。

```
-bash-3.2$ ls -l
合計 48
drwxr-xr-x 53 nakakuki user1 4096  8月 21  2010 2010_OLD
drwx-----  2 nakakuki user1   74  4月 11 10:30 Desktop
drwx-----  3 nakakuki user1   17  6月  6 16:06 JisshuB
drwx-----  3 nakakuki user1   17  6月 13 11:50 Joho2011
drwx-----  6 nakakuki user1   70  8月 21  2010 Mailboxes
drwx-----  6 nakakuki user1 4096  6月 15 16:26 Maildir
drwxr-xr-x 16 nakakuki user1 4096  6月  6 15:48 Windows
drwxr-xr-x  2 nakakuki user1   77  4月 14 14:23 public_html
-bash-3.2$ █
```

75

76 最初の drwxr-x-r-x というのは許可情報である。その次にオーナー、グループ、容量、  
77 変更日時、名前が出る。

78 新しいディレクトリを作成するときには次のようなコマンドを入力する。

79 mkdir *directoryname*

80 *directoryname* のところは適当な名前を付ける。mkdir はディレクトリを作るコマンド  
81 で、make directory の略である。GUI 側でも新しいディレクトリが出来ているのを確  
82 認してみよう。ディレクトリを消すには

83 rmdir *directoryname*

84 と入力する。なお、ディレクトリの中身が空でないと rmdir によりディレクトリを消す  
85 ことは出来ない。もし、ディレクトリを消してしまったら、もう一度ディレクトリを作  
86 っておこう。

87

## 88 6. ディレクトリの移動

89 ディレクトリを移動する

90 cd *directoryname*

91 と入力すると移動できる。ディレクトリを

92 pwd

93 と入れて確認する。1つ上の階層に戻るには、  
94 `cd ..`  
95 と入れる。  
96 `cd`  
97 あるいは  
98 `cd ~`  
99 と入力するとホームディレクトリへ戻る。“~”はホームディレクトリを意味する。  
100 “~nakakuki”とするとアカウント nakakuki のホームディレクトリという意味になる。戻  
101 ってしまったら、再び先ほど作ったディレクトリのなかへ入っておく。

102

### 103 7. エディタの起動と強制終了

104 `gedit &`

105 と入力して GNOME Editor をバックグラウンドプロセスとして起動する。

106 `ps`

107 と入力するとプロセス番号(今後 PID という)が確認できる。

```
-bash-3.2$ gedit &  
[1] 31499  
-bash-3.2$ ps  
  PID TTY          TIME CMD  
28177 pts/7    00:00:00 bash  
31499 pts/7    00:00:00 gedit  
31503 pts/7    00:00:00 gconfd-2  
31520 pts/7    00:00:00 ps  
-bash-3.2$ kill -9 31499  
-bash-3.2$ █
```

108

109 `gedit` のプロセス番号を確認しよう。確認したら、110 `kill -9 PID`

111 と入れる。ここで、*PID* は実際のプロセス番号(プロセスの起動により変わる。ここでは  
112 31499)である。こうするとプロセスが強制終了する。アプリケーションが暴走したと  
113 きなどに使用する。

114

- 115 **まとめ**
- 116 CUI ではコマンドを入力して計算機を利用する。
- 117 アプリケーションを動かすときは名前を入力する。
- 118 コマンドの書き方は英語の命令文と同じ構文で書く
- 119
- 120 **Linux のコマンド**
- 121 カレントディレクトリの表示 `pwd`
- 122 ディレクトリの移動 `cd directoryname`  
(*directoryname* はディレクトリ名)
- 123
- 124 ホームディレクトリに戻る `cd ~` または `cd`
- 125 1つ上の階層のディレクトリに移動 `cd ..`
- 126 ディレクトリを作成する `mkdir directoryname`
- 127 ディレクトリの削除 `rmdir directoryname`
- 128 アプリケーションのフォアグラウンド起動 `firefox`
- 129 アプリケーションのバックグラウンド起動 `gedit &`
- 130 自作アプリケーション(実行可能ファイル)の起動
- 131 `./a.out`  
(または、コンパイル時に付けた名前)
- 132
- 133 フォアグラウンドプロセスの強制終了 `Ctrl+c`
- 134 バックグラウンドプロセスの強制終了 `kill -9 PID` (*PID* はプロセス番号)
- 135 端末で前に入力したコマンドを出す ↑ (上向きカーソルキー)
- 136 コマンドやファイル名を推測させる `tab` (`tab` キー)
- 137 コピー マウスのカーソルでなぞる
- 138 貼り付け マウスの中ボタン(ホイール)を押す
- 139 これら2つはエディタでも使える
- 140
- 141 **エディタのショートカット**
- 142 コピー (copy) `Ctrl+c`
- 143 切り取り (cut) `Ctrl+x`
- 144 貼り付け (paste) `Ctrl+v`
- 145 やり直し (undo) `Ctrl+s`
- 146 上書き保存 (save) `Ctrl+s`
- 147 Windows でも同じ。Mac は `Ctrl` の代わりにコマンドキー。

- 148 許可情報の意味
- 149 drwxr-xr-x
- 150 最初の d ファイルの種類。dがついているとディレクトリ、ファイルは無印-
- 151 左の rwx 本人に対する許可情報
- 152 r:読み取り可能, w:書き込み可能, x:実行可能
- 153 ディレクトリの場合は x がないと中へ入れない。
- 154 ファイルの時は実行可能ファイル(executable)を表す。
- 155 真ん中の r-x 本人以外の同グループユーザーに対する許可情報
- 156 この場合、書き込みのみ不可。
- 157 右の r-x 他のグループのユーザーに対する許可情報
- 158 この場合、書き込みのみ不可。
- 159 -rwx-----となっていれば、本人のみ読み書き可能な実行可能ファイル
- 160
- 161 ディレクトリ構造のイメージ
- 162 たくさん引き出しのあるロッカー→引き出しの1つ1つに名前→引き出しの中に仕切
- 163 りがあって一つ一つに名前→仕切りの中にボックス(フォルダ)があって1つ1つに名
- 164 前→引き出しの中にバイнда(ファイル)があって1冊1冊に名前
- 165 最初の引き出しには単に"/"という名前がついている。これをルートディレクトリとい
- 166 う。引き出しや仕切りの1つ1つをディレクトリという。ファイルまでどのディレクト
- 167 リを通過していけばたどり着くかの道のりをパスと言う。/から名前を書いていけば、ど
- 168 のファイルまでもたどり着く道のりを示せる。これを絶対パスという。それ以外を相対
- 169 パスという。
- 170
- 171

- 172 付録
- 173 **Linux のスクリーンショットの取り方**
- 174 アクセサリ→スクリーンショットの取得を開く
- 175 ウィンドウのみを選ぶ
- 176 時間を 1～数秒後などにして、スクリーンショットボタンを押す。
- 177 スクリーンショットを撮りたいウィンドウをアクティブにする。
- 178 1～数秒後に保存ダイアログが出るので、画像を保存する(フォーマットは PNG)
- 179