

# 情報通信工学演習 I

国島丈生

2003-06-10

## 1 今回の概要

### 1.1 演習の内容

- UNIX コマンドによるファイル操作
- UNIX(所有者・グループ、権限)
- `rm -rf /`
- 片付け

### 1.2 レポート

以下の「演習」のうち「(レポート)」と書かれたものについてまとめよ。提出方法は次の通り。

- レポート内容を `~/public.html/report20030610.html` という HTML ファイルにまとめ、次週の演習開始時まで演習室の WWW サーバに転送しなさい。
- ページの最初に「情報通信工学演習 I レポート課題 (2003-06-10 分)」と書き (h1 エレメントを使用)、その次に自分の名前・学籍番号を書く (h2 エレメントを使用) こと。
- ページの最後に自分のメールアドレスを書く (address エレメントを使用) こと。

## 2 UNIX コマンドによるファイル操作

UNIX には第 1～3 週で学んだもの以外に多くのコマンドがあり、これらを組み合わせることでさまざまなことができるようになっている。ここでは、第 1～3 週で学んだコマンドなどを用いて、具体的なファイル操作を考えることを行う。

## 2.1 wc

### — 演習 1 —

wc というコマンドがどのようなものであるか、オンラインマニュアルで調べよ。Vine Linux では jman というコマンドで日本語マニュアルが参照できる。

### — 演習 2 —

前回の演習で作成した index.html の行数を求めるコマンドを示せ。

### — 演習 3(レポート) —

「/usr/bin というディレクトリに置かれていて、a で始まるファイルの数」を求めるには、どのようなコマンドを実行すればよいか。

## 2.2 grep

### — 演習 4 —

grep とはどのようなコマンドであるか、オンラインマニュアルで調べよ。

### — 演習 5 —

前回の演習で作成した ~/usrbinlist(/usr/bin に置かれているファイルの一覧)のうち、“zip” という文字列が含まれている行だけを入力するコマンドを示せ。

### — 演習 6(レポート) —

/usr/bin というディレクトリに置かれていて、“emacs” という文字列を含むファイルの一覧を入力するコマンドを示せ。作業ファイルを作らない方法を考えること。(演習室で行う場合には、“emacs” の代わりに “zip” を用いる)

### — 演習 7(レポート) —

/usr/bin というディレクトリに置かれていて、“emacs” という文字列を含むファイルの数を出力するコマンドを示せ。作業ファイルを作らない方法を考えること。(演習室で行う場合には、“emacs” の代わりに “zip” を用いる)

## 2.3 sort

### — 演習 8 —

sort とはどういうコマンドであるか、オンラインマニュアルで調べよ。

#### — 演習 9 —

/etc に置かれているファイルのうち、.conf という文字列で終わるものについて、ファイル名と文字数の一覧を出力するコマンドを示せ。(ヒント: wc を使う。出力形式は wc の出力そのままでよい。また全ファイルの合計文字数が表示されてもかまわない)

#### — 演習 10(レポート) —

演習 9 の結果を、文字数が小さい順に整列して表示する方法を示せ。作業ファイルを作らないこと。

#### — 演習 11(レポート) —

演習 9 の結果を、文字数が大きい順に整列して表示する方法を示せ。作業ファイルを作らないこと。また、全ファイルの合計文字数が一番最初に表示されてもかまわない。

## 3 UNIX における権限 (パーミッション)

### 3.1 読み出し権限

前回作成した一般ユーザでログインし、以下の演習を行え。

#### — 演習 12(レポート) —

前回の演習 16 で作成した ~/usrbinlist のパーミッションはどのようなになっているか。どのようにして調べたか。

#### — 演習 13(レポート) —

~/usrbinlist を、すべてのユーザ (所有者を含む) から読み出しができないようにパーミッションを変更せよ。その方法を示せ。

#### — 演習 14(レポート) —

現在の ~/usrbinlist の内容を more で表示しようとするとなんが起ころか示せ。

#### — 演習 15 —

~/usrbinlist のパーミッションを、現在の状態から所有者のみ読み出しができるように変更せよ。

#### — 演習 16(レポート) —

現在の ~/usrbinlist の内容を more で表示しようとするとなんが起ころか示せ。

### 3.2 ユーザと権限

ここまでとは別の一般ユーザでログインしなおし、引き続き以下の演習を行え。

#### — 演習 17 —

演習 12 から演習 16 までを行ったユーザのホームディレクトリに移動せよ。

#### — 演習 18(レポート) —

`groups` というコマンドを実行し、今ログインしているユーザがどのグループに属しているか調べよ。

#### — 演習 19(レポート) —

演習 8 のディレクトリにある `usrbinlist` の内容を `more` で表示しようとする何が起こるか示せ。なぜこのようなことが起きるのか、理由を述べよ。

#### — 演習 20(レポート) —

演習 10 のディレクトリにある `usrbinlist` のパーミッションを、すべてのユーザから読み出し可能にしようと試みよ。何が起こったか。パーミッションはどうなったか。

### 3.3 特権ユーザ root

次に `root` というユーザでログインしなおし、引き続き以下の演習を行え。

#### — 演習 21 —

演習 1 から演習 6 までを行ったユーザのホームディレクトリに移動せよ。

#### — 演習 22(レポート) —

`groups` というコマンドを実行し、今ログインしているユーザがどのグループに属しているか調べよ。

#### — 演習 23(レポート) —

演習 11 のディレクトリにある `usrbinlist` の内容を `more` で表示しようとする何が起こるか示せ。

#### — 演習 24(レポート) —

演習 11 のディレクトリにある `usrbinlist` のパーミッションを、すべてのユーザから読み出し可能にしようと試みよ。何が起こったか。パーミッションはどうなったか。

### 3.4 実行権限

あらためて一般ユーザでログインし、次の演習を行え。

#### — 演習 25 —

emacs で次のような内容のファイルを作成し、lsbackup という名前で保存せよ。これは、emacs でファイルを編集したときに自動的にできるバックアップファイル (ファイルの末尾に~がついたファイル) の一覧を出力するものである。

```
#!/bin/sh
ls *~
```

ここで作成した lsbackup はシェルスクリプトと呼ばれ、2 行目以降に書かれたコマンドを連続して実行することを示す。いわば、UNIX のコマンドを使って、ある種のプログラムを記述しているものだといふことができる。本演習では簡単なものしか紹介しないが、かなり複雑な処理も記述することができるので、興味のある方は自習されたい。

いったん作成したシェルスクリプトは、ls などのシステム標準のコマンドと同様に、コマンドとして実行することができる。ただし、実行するには、そのシェルスクリプトに実行権限を与えておく必要がある。

#### — 演習 26(レポート) —

~/public\_html というディレクトリに移動し、演習 15 で作成した lsbackup を実行してみよ。実行は、ターミナルエミュレータ上で次のようにする (% はプロンプトであることに注意)。何が起こったか。

```
% ~/lsbackup
```

#### — 演習 27 —

~/lsbackup のパーミッションを調べよ。

#### — 演習 28 —

~/lsbackup に対して、誰でも実行できるようにパーミッションを変更せよ。

#### — 演習 29(レポート) —

~/public\_html というディレクトリに移動し、演習 15 で作成した lsbackup を実行してみよ。結果を示せ。

シェルスクリプトの応用として、 $\text{\LaTeX}$  を実行したときに作成される \*.dvi, \*.aux, \*.log ファイルをまとめて消去する、といったものが考えられる。ここまでの演習の知識で作成することができるので、興味のある方は試みられたい。

## 4 rm -rf /

ユーザ root で “rm -rf /” を実行する実験を行う。

このコマンドの意味は「/というディレクトリを、その中にあるファイルやディレクトリも含めてすべて (-r オプション)、強制的に (-f オプション) 消去せよ」ということである。

UNIX のファイルシステムは/からはじまる木構造になっている。また、root は強力な権限を持つユーザであり、すべてのファイルやディレクトリを変更・上書き・消去できる権限を持つ。一方、

UNIX では、今実行しようとする `rm` や、各種 `daemon` プログラム、また UNIX の実行の核であるカーネル (kernel) とよばれる実行ファイルもすべて、一つのファイルとして存在している。

したがって、この実験の目的は、「UNIX で、現在実行中のプログラムのファイルを消去しようとするとなぜ起こるか確かめる」と言い換えることができる。

#### —— 演習 30(レポート) ——

この実験について、以下の内容をまとめ報告せよ。

1. 実験結果の予想。かならず実験を始める前に各自で結果の予想を立てよ。
2. 実験結果の報告 (予備実験を含む)。(途中での試行結果、最後にどうなったか)
3. 実験結果の考察。予想と合っていたかどうか、なぜこのような結果が出て、自分の予想はどこが違っていたのか、など。

実際に何が起こるか、結果を記した WWW ページがある (URL は演習中に指示)。このページを見たり、書籍を参照するなどしてまとめること。

以降の手順を始めると、システムは復旧できなくなる。レポートを作成するのに必要な資料・データが得られているか、充分確認すること。

ユーザ `root` でログインし、まず以下の予備実験を行う。

- `emacs`、`mozilla`、ターミナルエミュレータを 1 つずつ起動しておく。
- `/usr/bin/emacs`、`/usr/bin/mozilla` を `rm` コマンドを用いて消去する。
- `/usr/bin/emacs` を消去した後、`emacs` や `mozilla` が新たに起動できるか。
- `/usr/bin/emacs` を消去した後、あらかじめ起動しておいた `emacs` でファイルが新たに作成できるか。
- `/usr/bin/mozilla` を消去した後、あらかじめ起動しておいた `mozilla` で WWW が閲覧できるか。

この後、ターミナルエミュレータ上で、以下のコマンドを実行する (`#` はプロンプトであることに注意)。最後の `&` を忘れないこと。

```
# rm -rf / &
```

実行中はハードディスクへのアクセスランプが激しく点灯する。点灯が終わったら実行が終了したと考えてよい。実行時間は OS によって大きく異なるが、今回使用している Vine Linux では 5 分程度で終了するようである。

## 5 片付け

次のグループの実験のために、組み立てた PC を元の状態に分解し、片付ける。部品を壊さないよう、丁寧に扱うこと。また、元の箱・袋に入っているか、ケースのネジの締め忘れがないか、などにも注意しておく。