

# アセンブリ言語とオペレーティングシステムの実際

樋口さぶろお

龍谷大学工学部数理情報学科

情報処理の基礎 L13(2015-01-14 Wed)

## 今日の目標

- コンパイラが高級言語を機械語に変換する手順を説明できる
- 変換の各段階のファイルがビットパターンとしてどのように記録されているか説明できる
- OS上でどのようなプロセスが実行されているか観察できる



<http://hig3.net>

## L12-S1

Quiz 略解+コメント:オペレーティングシステム

略解

4.

OS の API, ライブラリが違うから普通は動作しない.

コメント

OS の間のこの違いを吸収するために, さらにもう一段の仮想化, OS の仮想化を行うことも行われている

## L12-S2

Quiz 略解+コメント:マルチタスキング

略解

3.

コメント

これは 1 個の AP の中の話.

## L12-S4

## Quiz 略解+コメント:OS の機能

## 略解

## 4.

## コメント

- ① プロセスは他のプロセスの仮想記憶にアクセスできないように OS が管理している
- ② Control-C でシェルを通じて OS に指示を出すことにより, 次のマルチタスク切り替え機会には, そのプログラムに切り替えないようにすることができる.
- ③ シェルに指示を出して, 好きなタイミングで AP の実行を始められる.
- ④ それは, Linux の C でも Windows の C でもそうで, C 言語の仕様が一つに定まっているから. OS の機能の話ではない.

## 準備

- チーム別座席指定
- Linux にログインします.
- Firefox ブラウザを起動して, `http://hig3.net` → RaMMoodle → 情報処理の基礎 (予習問題をやるところ) を開いておきましょう.

## ここまで来たよ

- 1 略解:オペレーティングシステム
- 2 アセンブリ言語とオペレーティングシステムの実際
  - アセンブリ言語
  - プロセス
  - 文字列の符号化

## アセンブリ言語

C言語で書かれたプログラムは、アセンブリ言語、機械語に変換されて、タスク=プロセスとして実行されるのでした。これを観察してみましょう。次のプログラムを Emacs や gedit などのエディタで作成し、ファイル名 `fiptest.c` で、ホームなど自分のわかる位置に保存しましょう。細部は違っていいですが、日本語は使わないください。

```
#include <stdio.h>
int main(){
    int i;
    int n=0;
    while(1){
        i++;
        n=n+i;
        printf("%d\n",n);
    }
    return 0;
}
```

→ 提出物 1

## ふだんなら

```
cc fiptest.c  
./a.out
```

とすところですが、`cc fiptest.c` は実際は何個かの作業を一度に行っています。

C 言語 → ? → アセンブリ言語 → 機械語 → 実行可能プログラム  
途中段階で止めてみましょう。

第1段階まで省略

第2段階まで

```
cc -S fiptest.c
```

同じディレクトリにアセンブリ言語のプログラム `fiptest.s` ができているはずです。

```
emacs fiptest.s
```

やダブルクリックで、中身を観察してみましょう。

→ 先頭 10 行が提出物 2

## 第3段階まで

```
cc -c fiptest.c
```

機械語のプログラム `fiptest.o` ができているはずです。

```
xxd fiptest.o
```

として、どのようなビットパターン (16 進表示) になっているか観察してみましょう。などとして、中身を観察してみましょう。

→ 先頭 10 行が提出物 3

出力は 3 列にわかれますが、アドレス、ビットパターンの 16 進法表示、ビットパターンを文字列の符号化とみなして文字列にもどしたもので、は 16 進法でなくビットパターンを 2 進法そのままに表示します。

```
xxd -b fiptest.o
```

### Quiz L13-1

`fiptest.c`, `fiptest.s`, `fiptest.o` の 3 つのファイルの先頭 10 行以上を貼り付けて提出してください。



## ここまで来たよ

- 1 略解:オペレーティングシステム
- 2 アセンブリ言語とオペレーティングシステムの実際
  - アセンブリ言語
  - プロセス
  - 文字列の符号化

## プロセス

ユーザがシェルを通じて OS にプログラムの実行を指示すると、プログラムの隔離された実行環境である **プロセス** が作られるのでした。プロセスを観察しましょう。

### 準備

メニューから

- システム > システム管理 > システム・モニタ を起動しましょう。
- アプリケーション > アクセサリ > 端末 を 2 個起動しましょう。

ps コマンドで観察端末で、

```
ps -F
```

と入力してみましょう。この端末に関係するプロセスが表示されます。

```
ps -eF
```

と入力してみましょう。OS 上のすべてのプロセスが表示されます。

```
ps -F -u ユーザ名
```

と入力してみましょう。ユーザ名=学籍番号です。ユーザのプロセスが表示されます。

端末で、

```
oclock &  
xclock &
```

(& も入力する) とした後、

```
ps -F
```

してみましょう。

## システム・モニタで観察

今度は、システム・モニタのリソースを観察しながらやってみましょう。

```
cc fiptest.c
```

してみます。

システム・モニタのリソースとプロセスを観察しながらやってみましょう。

```
./a.out
```

## Quiz L13-2

oclock, xclock, ./a.out がそれぞれメモリをどれだけ使っているか、システム・モニタから読み取って教えてください。

## ここまで来たよ

- 1 略解:オペレーティングシステム
- 2 **アセンブリ言語とオペレーティングシステムの実際**
  - アセンブリ言語
  - プロセス
  - **文字列の符号化**

## 文字列の符号化 I

Emacs で、半角英数文字だけで自分のローマ字の名前を書いたファイルを作り、name.txt として保存しましょう。

以下、端末で。

文字数を数えるコマンド

```
wc name.txt  
wc fiptest.c
```

これらの情報量がどれだけでしょうか？ 英数字 1 文字は 1B=8bit で符号化されるのでした。確かめてみましょう。

```
ls -l name.txt  
ls -l fiptest.c
```

または。

メニュー > **場所 > ホーム・フォルダ** で、右クリック > プロパティ  
どのように符号化されているのでしょうか？

## 文字列の符号化 II

```
xxd name.txt  
xxd fiptest.c  
xxd fiptest.s  
xxd fiptest.o
```

### QuizL13-3

ASCII コードの表と、`xxd name.txt` を参照して、自分の名前 (ローマ字) が 16 進法でどのように符号化されるか答えよう。

## ASCIIコード表 (0-127) L05 より再録

| 10進 | 16進 | 文字  |    |    |     |    |    |    |    |    |   |
|-----|-----|-----|----|----|-----|----|----|----|----|----|---|
| 0   | 00  | NUL | 16 | 10 | DLE | 32 | 20 | SP | 48 | 30 | 0 |
| 1   | 01  | SOH | 17 | 11 | DC1 | 33 | 21 | !  | 49 | 31 | 1 |
| 2   | 02  | STX | 18 | 12 | DC2 | 34 | 22 | "  | 50 | 32 | 2 |
| 3   | 03  | ETX | 19 | 13 | DC3 | 35 | 23 | #  | 51 | 33 | 3 |
| 4   | 04  | EOT | 20 | 14 | DC4 | 36 | 24 | \$ | 52 | 34 | 4 |
| 5   | 05  | ENQ | 21 | 15 | NAK | 37 | 25 | %  | 53 | 35 | 5 |
| 6   | 06  | ACK | 22 | 16 | SYN | 38 | 26 | &  | 54 | 36 | 6 |
| 7   | 07  | BEL | 23 | 17 | ETB | 39 | 27 | '  | 55 | 37 | 7 |
| 8   | 08  | BS  | 24 | 18 | CAN | 40 | 28 | (  | 56 | 38 | 8 |
| 9   | 09  | HT  | 25 | 19 | EM  | 41 | 29 | )  | 57 | 39 | 9 |
| 10  | 0A  | LF  | 26 | 1A | SUB | 42 | 2A | *  | 58 | 3A | : |
| 11  | 0B  | VT  | 27 | 1B | ESC | 43 | 2B | +  | 59 | 3B | ; |
| 12  | 0C  | FF  | 28 | 1C | FS  | 44 | 2C | ,  | 60 | 3C | < |
| 13  | 0D  | CR  | 29 | 1D | GS  | 45 | 2D | -  | 61 | 3D | = |
| 14  | 0E  | SO  | 30 | 1E | RS  | 46 | 2E | .  | 62 | 3E | > |
| 15  | 0F  | SI  | 31 | 1F | US  | 47 | 2F | /  | 63 | 3F | ? |



|    |    |   |    |    |   |     |    |   |     |    |     |
|----|----|---|----|----|---|-----|----|---|-----|----|-----|
| 64 | 40 | @ | 80 | 50 | P | 96  | 60 | ' | 112 | 70 | p   |
| 65 | 41 | A | 81 | 51 | Q | 97  | 61 | a | 113 | 71 | q   |
| 66 | 42 | B | 82 | 52 | R | 98  | 62 | b | 114 | 72 | r   |
| 67 | 43 | C | 83 | 53 | S | 99  | 63 | c | 115 | 73 | s   |
| 68 | 44 | D | 84 | 54 | T | 100 | 64 | d | 116 | 74 | t   |
| 69 | 45 | E | 85 | 55 | U | 101 | 65 | e | 117 | 75 | u   |
| 70 | 46 | F | 86 | 56 | V | 102 | 66 | f | 118 | 76 | v   |
| 71 | 47 | G | 87 | 57 | W | 103 | 67 | g | 119 | 77 | w   |
| 72 | 48 | H | 88 | 58 | X | 104 | 68 | h | 120 | 78 | x   |
| 73 | 49 | I | 89 | 59 | Y | 105 | 69 | i | 121 | 79 | y   |
| 74 | 4A | J | 90 | 5A | Z | 106 | 6A | j | 122 | 7A | z   |
| 75 | 4B | K | 91 | 5B | [ | 107 | 6B | k | 123 | 7B | {   |
| 76 | 4C | L | 92 | 5C | \ | 108 | 6C | l | 124 | 7C |     |
| 77 | 4D | M | 93 | 5D | ] | 109 | 6D | m | 125 | 7D | }   |
| 78 | 4E | N | 94 | 5E | ^ | 110 | 6E | n | 126 | 7E | ~   |
| 79 | 4F | O | 95 | 5F | _ | 111 | 6F | o | 127 | 7F | DEL |

## Maple T.A.

### 答の入力方法

C言語みたいな感じ. 半角英数字だけで入力してください.

$\sin x \rightarrow \sin(x)$

$x^2y \rightarrow x \ x \ y$  (スペースに注意) または  $x*x*y$

ただし,

$x^n \rightarrow \text{pow}(x,n) \quad x^{\wedge}n$

$\sin^2 x \rightarrow (\sin(x))^{\wedge}2$

^ キーを押すと, 表示されませんが上付き入力モードに入ります. 上付き入力を終了するには, Control と / の同時押し.

$x^ny$  は (コンマは不要です),  $x$ ,  $\wedge$ ,  $n$ , Control+/,  $*$ ,  $y$  とタイプ.

## 連絡

- 予習問題, 非参照 Quiz は今後ありません.
- 2015-01-14 水54 たぶん **プチテストリベンジ** 参加受付終了. プチテストと同じ出題計画. これと前回のプチテストのうち, 点数の高いほうを30ピーナッツ分とします. プチテストで70%以上を得た人は, 無視してファイナルトライアルに集中することをおすすめします.
- 2015-01-15 木 ころまで. **レポート問題提出期限** RaMMoodle から個人別問題取得.
- 2015-01-21 水3 補講. 教室でなくても, この日時でなくてもかまいません. 受講期限 2015-01-27 火
- 2015-01-28 水 たぶんファイナルトライアル (40ピーナッツ) 出題計画は 2015-01-21 ころまでに更新して補講の際に示します.
- 樋口のオフィスアワー 木6(1-539), 金昼(7-002/1-502).
- チューターは **月**火水木昼 (1-614).