

情報工学実験 1 課題 2 アセンブリ言語

担当教員 — 株式会社あいはら 研究開発チーム 山田 泰司 <taiji@aihara.co.jp>

教員紹介

- ・ 東京理科大学 工学部 情報工学科「メディアデザイン」(3年次)で iPhone モバイルプログラミングを指導 (2018 ~)
- ・ 埼玉大学 工学部 情報工学科「実践的システム開発Ⅱ」(3年次)で iPhone モバイルプログラミング指導を担当 (2017 ~)
- ・ 埼玉大学 工学部 情報工学科「情報システム工学Ⅰ」(2006 ~ 2016)、「情報システム工学Ⅱ」(2008 ~ 2016) C によるグラフィックス (Xlib, OpenGL, SDL etc.)、RPC (Remote Procedure Call)、ソケットプログラミング、マルチスレッド、並列プログラミングを指導
- ・ C/C++, /bin/ksh, /usr/bin/sed, Javascript その他よろず技術系プログラミングとシステム設計・開発にずっと従事

アセンブラで「カッコイイ」「凄い」「役立った」「使える」「使われている」なっとなこと

1. かつてのドラクエⅢ作者の内藤寛氏が Youtube 実況で 34 年越しのバグを修正実況！[ハンドアセンブリ！\(動画\)](#) & [アセンブリ！\(動画の続編\)](#)
2. NASA の 1977 年出発した探査機ボイジャーの不具合を制御プログラムを遠隔でプログラマが[アセンブリ言語](#)で修正！
3. Boost C++ Template Library の uBLAS は「Expression Template」という技法で、例えば、ある特性を持った行列の積の計算で「打ち消しあって零」などの評価が「コンパイル時」に完了してしまう。確かめるためには、コンパイルされた機械語を[逆アセンブリ](#)すれば、アセンブリ言語がわかる人にはわかる。
4. LaTeX の基本システム TeX 作者の Donald E. Knuth は名著「The Art of Computer Programming」(俗に TAOCP) にて機械語 [MIX](#)、さらに現代的な機械語 [MMIX](#) を定義している。これはプログラミング学習のための仮想マシンであり、このことから「機械語を理解することはプログラミングを理解すること」に繋がることがわかる。
5. macOS の strcpy は素直な実装より高速 → [実稼働しているソースコード](#) https://opensource.apple.com/source/Libc/Libc-997.90.3/x86_64/string/strcpy.s.auto.html なんとアセンブラで書かれている！(SIMD 命令を利用)

課題2 — アセンブリ言語

サポートページの URL(s) について

LETUS にも書いてありますが、以下のサイトが重要です。

メインサーバ URL

- <http://www.hisenkei.net/~sima/lecture/itex1/>
ID: itex12 PASSWORD: asm2021

ミラーサーバ URL (メインサーバが不調のとき)

- <http://www.aihara.co.jp/~taiji/lecture/ITex1/>
ID: なし PASSWORD: なし

必要に応じて対面でも Zoom (再び必要になる事態や事情がある方など)

- zoom URL Zoom ID: 93424277030

情報工学全般 — 拙作の補助教材

アセンブリ言語に先立ち、エディタ Emacs、端末など
手前味噌ですが、超オススメです！

<http://www.aihara.co.jp/~taiji/lecture/>

シェルスクリプト、テキスト処理（正規表現）、Emacs、TeX、Javascript

コンピュータ・アーキテクチャ、US/JIS 配列キーボード、Unix、

高校数学とアルゴリズム、数学全般、

プログラマ・リテラシー、

CPU の発展とプログラミング言語、などなど多数

実験・講義とは別に「一生モノの教養」として参考に見てみてください。

DOSBox — 諸問題への対策

画面が小さ過ぎ！ 「^ ¥ @ :」 キーが打てない！

DOSBox はフォーク DOSBox-X と比較して決して機能過多ではなく、**教育的には良い選択です**。しかし、以下の問題があり、作業に支障があるかもしれません。

1. 画面がリサイズ出来ず、昨今の高解像度モニタでは画面が小さ過ぎる。
2. 日本で一般的な JIS 配列キーボード (JPI06) に未対応で、打鍵できないキーがある。

以下のページでこの二つの問題の対策案を示しているので参考にして下さい。

1. [DOSBox ウィンドウサイズの変更方法について \(PDF 形式\)](#)
2. [DOSBox キーマッピングの変更方法について \(PDF 形式\)](#)

ここでは経緯上、対策案 A と B が示してありますが、A は読み流して B を実施して問題なければそれで良いと思います。さらに DOSBox-X についても言及しています。DOSBox-X は Linux 環境でインストール済みです。

nasm on macOS via Homebrew に対する注意

Apple Silicon (M1/M2) Mac では通すべき実行パス（後述）が異なります！

Homebrew とは macOS 用のパッケージ管理ツールです。GNU/Linux や Windows Subsystem for Linux (WSL) においては Ubuntu の `dpkg` や Fedora の `Dandified Yum`、BSD Unix においては FreeBSD の `port`、NetBSD の `pkgsrc` に相当します。macOS には Homebrew の他に MacPorts というパッケージ管理ツールもよく使われます。

Homebrew の問題点 ～「**混ざるぞ危険！**」

Homebrew for Intel Mac ではインストール先が `/usr/local` なので `/usr/local/bin` に実行パスがあることが問題です。何故なら、ソースコードのビルドシステム `configure` (GNU Autotools) はここを見て依存するので、他のパッケージ管理システムや `configure` による自前ビルドが、意図せず Homebrew に依存してしまうからです。Homebrew が歴史的に「**/usr/local** を占拠」したことは大きな判断ミスです。そもそも **Unix の「ファイルシステム階層標準 (FHS)」**にて **/usr/local** は局所的な用途に定義されているので、このような **/usr/local** の使用は誤りなのです。よって、**筆者は /Users/Homebrew にインストールされるように工夫しています**（但し、Homebrew 本家想定外ゆえ多少の不具合はあるかもしれません）。特に MacPorts も併用したい方は、以下の「非公式『筆者推奨』インストール方法」を参考にしてください。

- **Homebrew 非公式『筆者推奨』インストール方法** … https://www.aihara.co.jp/~taiji/unix-tips/darwin/#darwin_Homebrew

これに従えば Intel Mac でも M1 Mac でも `/Users/Homebrew/bin` にパスを通せば便利に使えます。また、**ログイン時に /Users/Homebrew/bin に実行パスを通しておきたければ、`/etc/paths.d/Homebrew` をエディタで新規作成し、このパスを書いておきましょう。**

nasm on macOS via Homebrew に対する注意

Apple Silicon (M1/M2) Mac では通すべき実行パス（後述）が異なります！（の続き）

インストール済みの方は以下の実行パスであることに今一度確認しましょう。

- Homebrew 公式 Intel Mac — /usr/local/bin
- Homebrew 公式 Apple Silicon Mac — /opt/homebrew/bin
- Homebrew 非公式『筆者推奨』インストール方法 — /Users/Homebrew/bin

そして、例えば前述サイトの拙作 `install-Users_Homebrew.sh` の場合、

```
% ./install-Users_Homebrew.sh
      (中略)
% sudo xcode-select --install
% /Users/Homebrew/bin/brew tap homebrew/cask
% /Users/Homebrew/bin/brew update
```

以上を実行して下さい。そして、nasm をインストールするには以下のようにします。フルパスは状況に応じて読み替えること。

```
% /Users/Homebrew/bin/brew install nasm
```

以上でホストコンピュータ側で nasm が以下のように使えるようになります。

```
% /Users/Homebrew/bin/nasm --help | less
```

実行パス in DOS, Windows, POSIX(macOS/Linux) or BSD Unix

それは打鍵したコマンドを探すパスのリスト — 環境変数 PATH

- まず、筆者の普段使いの macOS の環境変数 PATH をみてみましょう！

```
% echo $PATH          # 以下「:」コロン区切りのリスト。
```

```
/opt/local/bin:/opt/local/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin:~
```

このリストから実行すべきコマンド名を探すのです。では、

- コマンドを探すのはどなたでしょうか？ 答え：
 - macOS/Linux/BSD Unix — /bin/zsh または /bin/bash などのシェル
 - Windows PowerShell — PowerShell.exe
 - Windows コマンドプロンプト — cmd.exe
 - MS-DOS/DOSBox — COMMAND.COM (注) 同名だが別実装

実行パス in DOS, Windows, POSIX(macOS/Linux) or BSD Unix

それは打鍵した**コマンドを探すパス**のリスト — 環境変数 PATH (の続き)

カレントディレクトリに作った「**test.com/text.exe/test**」が実行できないのですが？

- かつて Unix でもカレントディレクトリ「**.**」に実行パスを通しておく慣習があり、第三者の tarball 内に (例えば) 「ls」というウイルスに感染させるコマンドが仕込まれてて、業界を騒がせた事件がありました。それ以降、Unix では実行パスにカレントディレクトリ「**.**」を通す習慣が排除されました。
- 一方、Windows ではコマンドライン・インターフェースの利用者が少ないことと過去の資産への影響との天秤で、カレントディレクトリに**暗黙に実行パスを通す**決まりは保持され続けています。
- しかし、過去の資産とは無縁な Windows PowerShell ではカレントディレクトリに実行パスは**暗黙には通さない**仕様となっています。よって、以下のような習慣をつければよいのです。

```
% ./test      # bash/zsh for Unix
> .%test      # COMMAND.COM/cmd.exe/PowerShell.exe for Windows in Japan
> .\test      # COMMAND.COM for DOSBOX
```

[豆知識] では Unix で `test` と打鍵した場合はどうなりますか？

/bin/test もしくはシェル組み込みコマンド test が実行されます (シェルスクリプトの「**[**」及び「**/bin/[**」の実体です)。

お知らせ

2023/04/24、05/15、06/05、06/26

- 次回以降のための準備（DOSBox 等）は3回目から開始します。
 1. Mac 向け環境整備の説明（GNU Emacs, DOSBox, nasm via Homebrew）
 2. Windows 向け環境整備の説明（GNU Emacs, DOSBox, nasm）
 3. Linux 向けは省略しますが、emacs, DOSBox-~~X~~, nasm がインストール済みです。
- では、皆さん、以上の「環境整備」を行なって下さい。