



プログラミング演習 1
CASL シミュレータ マニュアル



CASLシミュレータの概要

□ ブラウザ上で実行できる

- インターネットが使用できれば自宅でも実行できる

□ URL

- <https://www.officedaytime.com/dcaslj/>

シミュレータ画面

ソースリストの入力場所
(テキストエディタで作成した
ソースをコピペできる)

各種レジスタの内容

(メモリ上の) アドレス確認部

アセンブルオプション

CASLシミュレータ (CASL II 対応)

ソース

```
EX1      START
          OUT      OUTBUF,OUTLEN
          RET
OUTBUF  DC      'HELLO WORLD!'
OUTLEN  DC      12
          END
```

アセンブル

▶ || ⏪ ⏩ ⏴ ⏵

すべてのブレークポイントを削除 中止

16進 10進符号なし 10進符号付き

GR0 0000 GR1 0000 GR2 0000 GR3 0000
GR4 0000 GR5 0000 GR6 0000 GR7 0000
PR 000C SP FFFF ZF 0 SF 0 OF 0

アドレス(16進) 表示

| | | | | |
|------|------|------|------|------|
| 0000 | 7001 | 0000 | 7002 | 0000 |
| 0004 | 1210 | 0000 | 1220 | 0019 |
| 0008 | F000 | 0002 | 7120 | 7110 |
| 000C | 8100 | 0048 | 0045 | 004C |

以下は次回アセンブル時に有効になります

タブ位置 8 4

フォントサイズ(px)

実行開始ラベル

配置開始アドレス

SP初期値

ヘルプ CASL II 入門講座
動作確認: IE 6/7, Firefox 2, Opera 9
もっと使いやすいのがほしい方 → DCasI2をおすすめします。
Version 1.01 (C) Daytime 2007. All rights reserved.
daytime@officedaytime.com

コンソール

```
HELLO WORLD!
```

実行結果

| 0000 | EX1 | START | OUT | OUTBUF,OUTLEN |
|------|--------|-------|-----|----------------|
| 0000 | | OUT | | |
| 000C | | RET | | |
| 000D | OUTBUF | DC | | 'HELLO WORLD!' |
| 0019 | OUTLEN | DC | | 12 |
| | | END | | |

アセンブルリスト

ソースリストの作成

□ “ソース”欄に実行したいソースリストを入力

➤ おすすめの入力方法

1. (メモ帳などの) テキストエディタでソースコードを作成
2. 作成したソースコードをシミュレータにコピー&ペースト

```
1 EX1      START↓
2          OUT      OUTBUF,OUTLEN↓
3          RET↓
4 OUTBUF   DC      'HELLO WORLD!'↓
5 OUTLEN   DC      12↓
6          END[EOF]
```

コピー



```
ソース
EX1      START
          OUT      OUTBUF,OUTLEN
          RET
OUTBUF   DC      'HELLO WORLD!'
OUTLEN   DC      12
          END
アセンブル
```

ソースコードの実行1

□ ソース欄下部にある「アセンブル」ボタンをクリック

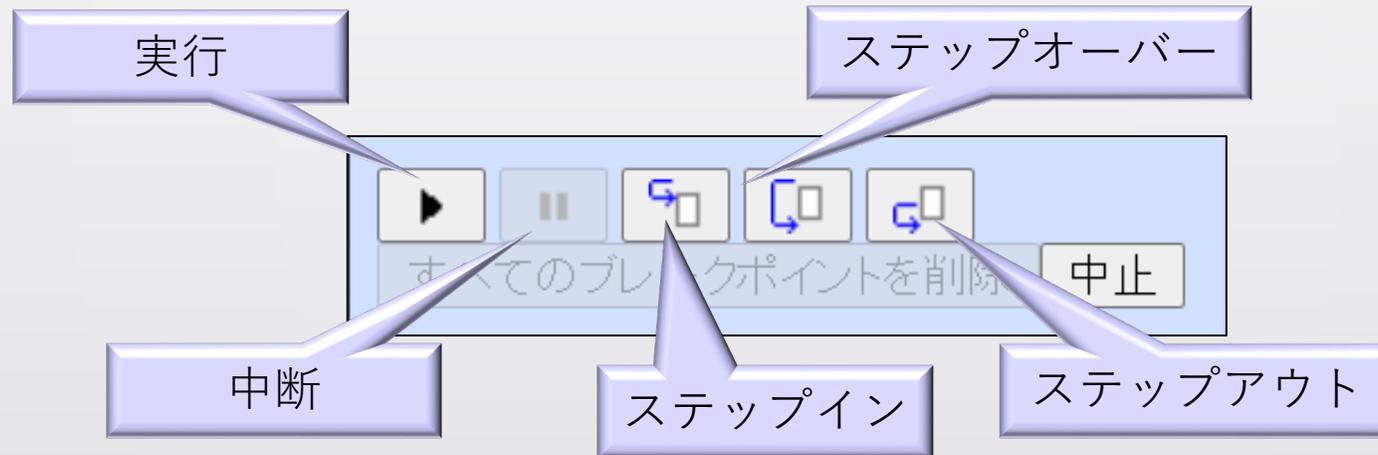
- エラーがあると・・・
 - 画面右上にエラー内容が表示
- エラーがないと・・・
 - 画面右下にアセンブルリストが表示
 - 実行開始行が黄色にハイライトされる

```
ソース
EX1          START
              OUT      RET      OUTBUF,OUTLEN
              RET
OUTBUF      DC      'HELLO WORLD!'
OUTLEN      DC      12
              END
```

アセンブルボタン

ソースコードの実行2

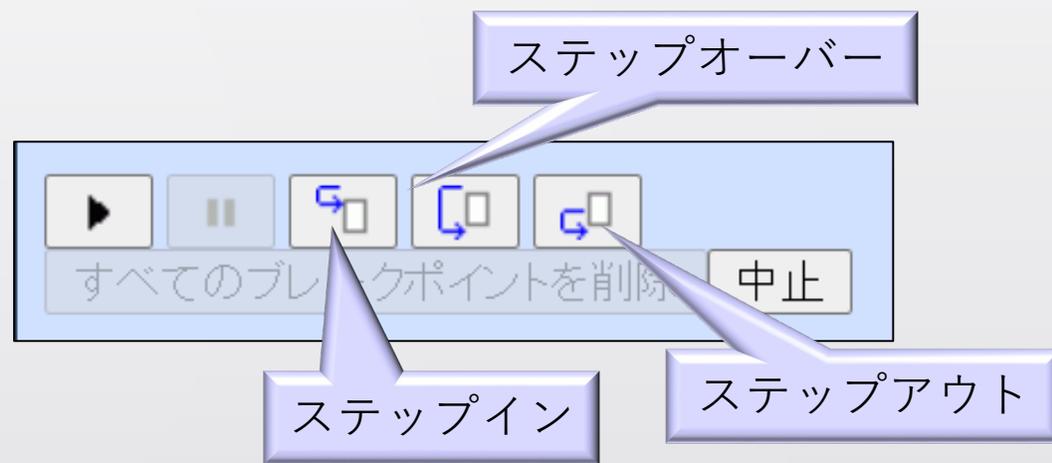
- エラーなくアセンブルできたら・・・実行ボタンをクリック



デバッグ機能

□ エラーなくアセンブルできるが，想定している実行結果が出力されないとき，デバッグしてみよう

- ステップイン：
アセンブルリストのハイライト行が1行進む
(レジスタの中身も更新)
- ステップオーバー：
サブルーチンないで停止しない
- ステップアウト：
サブルーチンから戻るまで実行後，停止

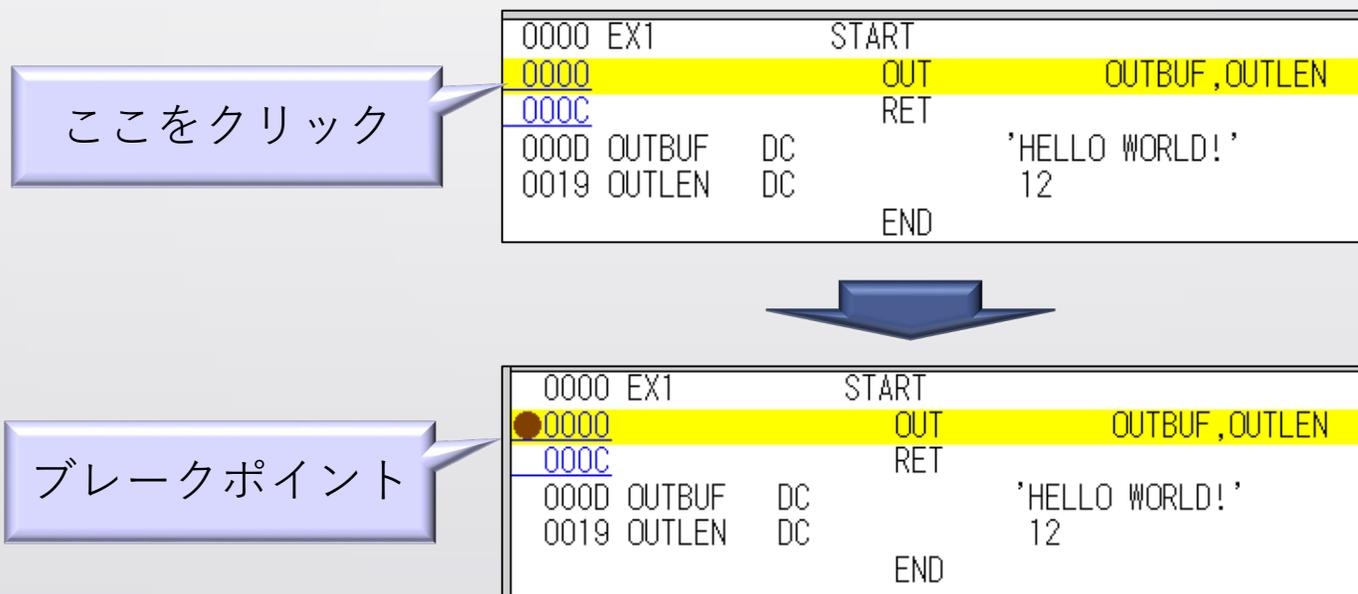


□ Visual Studioのようなステップ実行可能

デバッグ機能

□ ブレークポイント

➤ 任意の行で停止してほしいときに使用



レジスタ・メモリアドレスの確認方法

The screenshot shows a debugger interface with the following elements:

- 汎用レジスタ (GR0~GR7):** A callout box pointing to the General Purpose Registers (GR0-GR7) section.
- プログラムレジスタ:** A callout box pointing to the Program Register (PR).
- スタックポインタ:** A callout box pointing to the Stack Pointer (SP).
- Flags:** A callout box pointing to the Zero Flag (ZF), Sign Flag (SF), and Overflow Flag (OF).
- Memory Display:** A callout box pointing to the memory address input field and the memory dump table.

16進 10進符号なし 10進符号付き

| | | | | | | | |
|-----|------|-----|------|-----|------|-----|------|
| GR0 | 0000 | GR1 | 0000 | GR2 | 0000 | GR3 | 0000 |
| GR4 | 0000 | GR5 | 0000 | GR6 | 0000 | GR7 | 0000 |
| PR | 0000 | SP | FFFE | ZF | 0 | SF | 0 |
| OF | 0 | | | | | | |

アドレス(16進) 0000 表示

| | | | | |
|------|------|------|------|------|
| 0000 | 7001 | 0000 | 7002 | 0000 |
| 0004 | 1210 | 000D | 1220 | 0019 |
| 0008 | F000 | 0002 | 7120 | 7110 |
| 000C | 8100 | 0048 | 0045 | 004C |

ZF: Zero Flag
SF: Sign Flag
OF: Overflow Flag

ここに確認したいアドレスを入力し、表示をクリックするとそのアドレスから**16**語が表示