# Asm09 アセンブラー

## 実行方法

コマンドプロンプトで実行します。

asm09 オプション ソースファイル名

### オプション

|  |  |
| --- | --- |
| -v2 | **バージョン2の機能を有効にする**   * **SCOPE~ENDSCOPEディレクティブ** * **ZEXTディレクティブ** * **ダイレクトアドレッシング自動判別** |

## コメント

セミコロンから行末まではコメントとなります。

## 定数

|  |  |
| --- | --- |
| 10進定数 | 1つ以上の0~9 |
| 16進定数 | 0~9で始まり0~9またはA~Fと末尾のH  または$で始まり0~9またはA~F |
| 文字定数 | シングルまたはダブルクォーテーションで囲まれた文字 |

## シンボル

[シンボルに使える文字はアルファベットと\_.?@です。2](mailto:シンボルに使える文字はアルファベットと_.?@です。2)文字目以後は数字も使えます。

## 演算子

以下の演算子を使用できます。

|  |  |
| --- | --- |
| 単項 | 二項 |
| |  |  | | --- | --- | | + | 正符号 | | - | 負符号 | | HIGH | 上位バイト | | LOW | 下位バイト | | NOT | 反転(1の補数) | | |  |  | | --- | --- | | + | 加算 | | - | 減算 | | \* | 乗算 | | / | 除算 | | MOD | 剰余 | | SHL | 左シフト | | SHR | 右シフト | | AND | 論理積 | | OR | 論理和 | | XOR | 排他的論理和 | |

優先順位は以下の通りです。

|  |  |
| --- | --- |
| 優先順位 | 演算子 |
| 1 | \*, /, MOD |
| 2 | +, - |
| 3 | SHL, SHR |
| 4 | AND |
| 5 | OR, XOR |

## 疑似命令

|  |  |
| --- | --- |
| INCLUDE | 他のファイルを挿入します。 |
| PUBLIC | シンボルを他のモジュールから参照可能にします。 |
| EXTERN EXT | 他のモジュールのシンボルを参照します。 |
| CSEG | この命令以後をコードセグメントに配置します。 |
| DSEG | この命令以後をデータセグメントに配置します。 |
| ZSEG | この命令以後をダイレクトページセグメントに配置します。 |
| EQU | 定数を定義します。 |
| DEFB DB | バイト定数配置します。 |
| DEFW DW | ワード定数配置します。 |
| DEFS DS | メモリ領域を確保します。 |

### -v2オプション指定時

|  |  |
| --- | --- |
| ZEXTERN ZEXT | 他のモジュールのシンボルをダイレクトページとして参照します。 |
| SCOPE~ENDSCOPE | ラベルの有効範囲を指定します。ただしPUBLIC宣言されたラベルを除きます。 |

## ダイレクトアドレッシングとエクステンドアドレッシングの使い分け

|  |  |
| --- | --- |
| ダイレクト | 1. オペランドの左に<を付ける 2. オペランドがバイト値 3. (-v2オプション指定時) ダイレクトページセグメントに配置されたシンボル 4. (-v2オプション指定時)ZEXTまたはZEXTRNで宣言されたシンボル |
| エクステンド | 1. オペランドの左に>を付ける 2. オペランドがワード値または未定 |

## 独自機能

### 相対ジャンプの置き換え

BRA等の相対ジャンプ先が範囲外の場合、LBRA等に置き換えられます。

### 複数命令

|で区切ることで、1行に複数の命令を記述できます。

## 構造化命令

### IF-ELSE-ENDIF

|  |  |
| --- | --- |
| 記述 | 機能 |
| IF 条件  ブロック1  ELSE  ブロック2  ENDIF | 条件が真の場合はブロック1、偽の場合はブロック2を実行します。ELSEブロックは省略可能です。 |

### DO-WHILE-WEND

|  |  |
| --- | --- |
| 記述 | 機能 |
| DO  ブロック1  WHILE 条件  ブロック2  WEND | ブロック1を実行し、条件が成立する限りブロック2を繰り返します。 |

### 条件

HI, LS, CC, HS, CS, LO, NE, EQ, VC, VS, PL, MI, GE, LT, GT, LE を記述できます。(意味はBHI等の分岐命令と同じ)

# LinkBE リンカー

## 実行方法

コマンドプロンプトで実行します。

linkbe 出力ファイル名 コードセグメントアドレス データセグメントアドレス ゼロページセグメントアドレス オブジェクトファイル名…

セグメントアドレスには、4000-4fff,2000 のように複数範囲を指定できます。

拡張子の指定により、出力ファイルの形式は以下のようになります。

|  |  |
| --- | --- |
| 拡張子 | 形式 |
| CMT | PC-8001エミュレーター用 |
| P6 | PC-6001エミュレーター用(ローダー付き) |
| MZT | MZエミュレーター用 |
| CAS | MSXエミュレーター用 |
| RAM | PASOPIAエミュレーター用RAMPAK2 |
| PRG | JR-100エミュレーター用 |
| CJR | JR-200エミュレーター用 |
| L3 | ベーシックマスターレベル3エミュレーター用 |
| T64 | Commodore64,VIC-20エミュレーター用 |
| C10 | TRS-80 MC-10およびCoCoエミュレーター用 |
| HEX | インテルHEXフォーマット |
| S | モトローラSレコードフォーマット |
| 上記以外 | コードセグメントの内容そのまま(アドレス情報なし) |