

プロセッサと機械語

2年次前期（第12回）

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

CASL II のまとめ

2

アセンブリ言語CASL II

（教科書 p.29）

• 命令の種類

- ◆ 機械命令 ... LD, ST, ADDA, JP, RET
- ◆ マクロ命令 ... IN, OUT, RPUSH, RPOP
- ◆ アセンブラ命令
 - プログラムの構造の指示 ... **START**, **END**
 - データ語生成の指示 ... DC, DS

• 文法 ... 1行に1命令ずつ記述する

- ◆ ラベル ... 番地の記号的表現
 - 先頭英字で始まる1～8文字の英数字（英字は大文字）
 - GR0, ..., GR7 はラベルとしては使用不可
 - 行の先頭（第1桁）から記述しなくてはならない
- ◆ 命令
 - ラベルの後に1文字以上の空白を置く
 - ラベルが無い場合は行の先頭（第1桁）を空けて第2桁以降に
 - 命令コードとオペランド間は1文字以上の空白を置く
 - 複数のオペランド間は空白を入れない
- ◆ 注釈 ... 行中の任意位置の「;」から行末まで

3

アセンブリ言語CASL II

（教科書 p.29）

• 文法（続）

- ◆ 機械命令の記述形式
 - op gr,adr[,xr]
 - op adr[,xr]
 - op gr1,gr2
 - op gr
 - op
- ◆ op（命令コード）... LD, ST, ADDA, etc.
- ◆ gr（汎用レジスタ）... GR0 ~ GR7
- ◆ xr（インデックスレジスタ）... GR1 ~ GR7
- ◆ adr
 - ラベル ... どこかの行で定義されていなくてはならない
 - 10進定数 ... 10進数字の列で、先頭に「-（マイナス）」があっても良い
-1 と 65535, -2 と 65534, ... は同じと見なす
 - 16進定数 ... 「#」の後に4桁の16進数字
 - リテラル ... 「=」の後に 10進定数, 16進定数または文字定数
それらの格納された番地を示す

4

アセンブリ言語CASL II

(教科書 p.29)

● 定数の表記

◆ 文字定数(文字列)(シングルクォート「'」で囲む)

'abc' ... abc という文字列

DC 'abc' ... abc という文字列を3語に格納

#0061
#0062
#0063

◆ リテラル(=)

「=」の後に10進定数, 16進定数, 文字定数(文字列), ~~ラベル~~を書くと, それらを格納(DC)した番地を生成する事が出来る.

例:

=10 ... DC 10 を自動的に挿入し, その番地を表す

LD GR0,=10	{	LD GR0,TEN	
	:	:	:
	TEN	DC 10	#000A
LD GR0,='b'	{	LD GR0,BIE	
	:	:	:
	BIE	DC 'b'	#0062

5

アセンブリ言語CASL II

- 演習 LITST ... リテラルや文字定数を使用し, どのようなコードが生成されるか確認せよ(実行もして見よ).

```
LITST  START
        LD  GR0,=1      ; DC 1 が自動挿入されるか?
        LD  GR1,='a'    ; DC 'a' が自動挿入されるか?
        LAD GR2,='bcde' ; DC 'bcde' が自動挿入されるか?
        ST  GR1,0,GR2    ; 'bcde' の置かれた番地の先頭に 'a' を格納
        RET
STR      DC  'abc+def'   ; 7語分になるか?
END
```

6

アセンブリ言語CASL II

● アセンブリ言語の有り難味は?

- － 機械語での数字を記号(mnemonic)で書ける

[例] 20350102 → ADDA GR3,DAT,GR5

- － 参照番地(ラベル位置)を自動計算してくれる

0100	20350102	ADDA GR3,DAT,GR5	0100	20350103
0101	64000103	JUMP LAB	0101	20330201
0102	00000001	DAT: #0001	0102	64000104
0103	XXXXXXXX	LAB:	0103	00000001
			0104	XXXXXXXX

↑
挿入
参照先番地
がずれる

7

CASLプログラミング

● ポイント総括1

◆ 要求仕様の確認

- 入力データ, 出力データの場所, レジスタ非破壊指定 等

◆ フローチャートの活用

- フローチャート → フローの一系列変形 → ループの構成吟味
→ 入出力データや一時データを表す変数とGR等との対応決定
→ コーディング → デバッグ

◆ ループの構成

- ループ終了条件と条件分岐命令の選択
- ループ回数の確認
- FRのセット確認

◆ 配列のアクセス

- インデックスレジスタ(GR1~GR7)の使用(0~要素数-1)
 - 「LD BUF,GRn」型 と 「LD 0,GRn」型

8

CASLプログラミング

●ポイント総括2

- ◆スタック命令の利用
 - サブルーチンでのレジスタ非破壊のためのGRの退避・復元
 - GR0はPUSHできないがPOPは出来る
 - データの並べ替えに利用可能
 - 出力したいデータの並びと逆順に計算結果が求まる場合など
- ◆入出力命令IN,OUTの正しい使用
 - IN BUF,LEN の BUF は常に 256語
 - BUFの各語の下位8ビットは「文字コード」, 上位8ビットは 0
- ◆その他
 - 科目HPの「CASL－Ⅱ ノウハウ集」を参照

プロセッサと機械語

追加演習

CASLプログラミング

- 演習 NUMNUM ... IN BUF,LEN 命令により, 文字コードを読み込み, これらの文字コードの内, 数字を表す文字(0~9)の個数を数えて RSLT番地に格納するプログラム NUMNUM のフローチャートを作成せよ.

CASLプログラミング

- 演習 NUMNUM ... IN BUF,LEN 命令により, 文字コードを読み込み, これらの文字コードの内, 数字を表す文字(0~9)の個数を数えて RSLT番地に格納するプログラム NUMNUM を作成せよ.

NUMNUM START ; IN命令で読み込んだ文字の内, 数字の個数を数える

IN BUF,LEN

XOR GR0,GR0 ; 0 → GR0 (S:数字の個数集計用)

XOR GR1,GR1 ; 0 → GR1 (i:ループカウンタ&インデックシング)

CPA GR1,LEN ; i = 文字数 ?

JZE LAST ; 文字数=0 の場合も有り得る

LOOP

1文字取り出し

文字コード-#0030

V < 0 ならスキップ

V > 9 ならスキップ

S+1→S

ループカウンタを+1

i = 文字数 ?

if ループカウンタ≠文字数 goto LOOP

SKIP

LAST ST GR0,RSLT ; 結果の格納

RET

RSLT DS 1 ; 個数の格納場所

LEN DS 1 ; 文字数格納場所

BUF DS 256 ; 入出力バッファ

END

期末テストの要領

★注意:

- ◆PCは、WCASL-Ⅱ、本科目ページのみ参照可。
(他アプリ、他ページ参照は不正行為と見なします)
- ◆教科書他、紙の資料は全て参照可。
- ◆提出は配布した紙で
- ◆時間はチャイムまで

★アドバイス:

- ◆問題は多めかも知れません。出来るところから取り組んで下さい。
- ◆計算問題は手計算でもWCASL-Ⅱ利用でも構いません。慣れていればWCASL-Ⅱ利用の方が計算が早いと思われます。
- ◆机が狭くなるため、問題用紙は小さくしています。解答は明瞭に記載して下さい。不明瞭だと採点出来ませんので。