

計算機言語講義資料 第6回 C言語 続き

大阪大学 大学院情報科学研究科
2015年
長谷川亨
t-hasegawa@ist.osaka-u.ac.jp

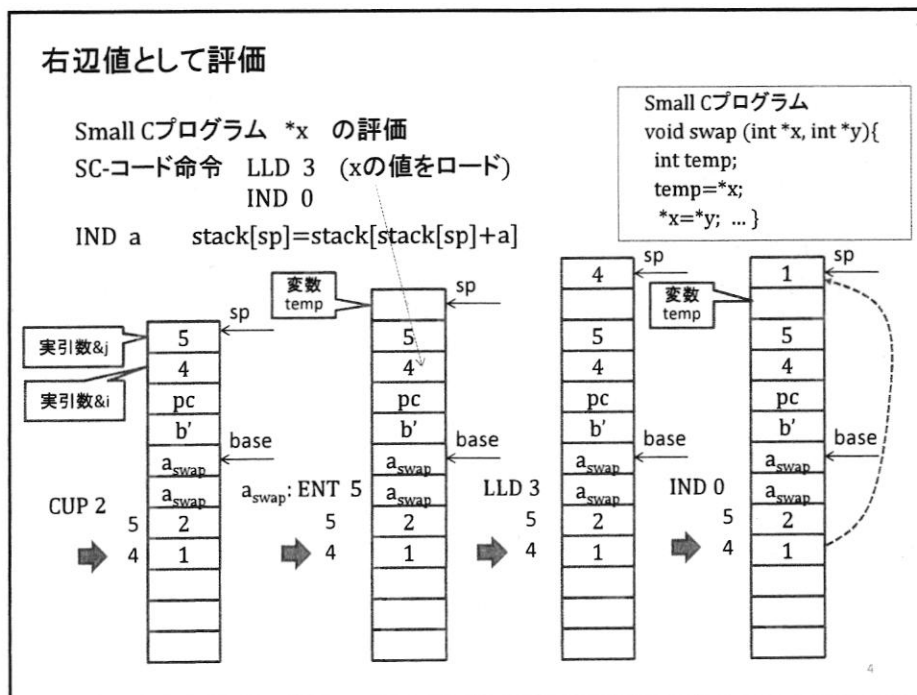
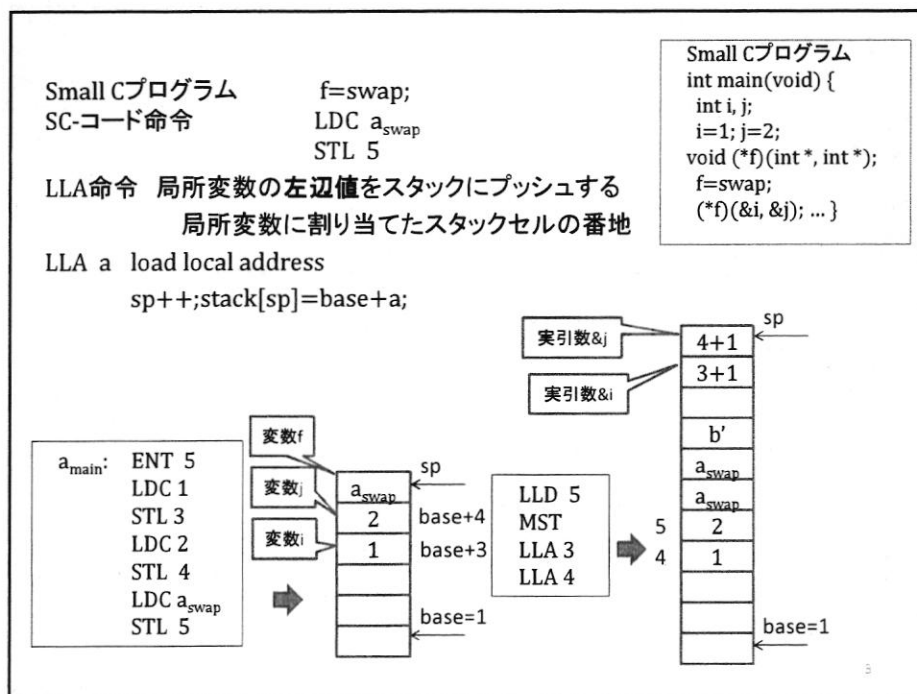
関数ポインタ、ポインタ変数

Small Cプログラム

```
void swap (int *x, int *y){
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
}
int main(void) {
    int i, j;
    void (*f)(int *, int *);
    i=1; j=2;
    f=swap;
    (*f)(&i, &j);
}
```

SC-コードプログラム

a _{main} :	ENT 5	a _{swap} :	ENT 5
	LDC 1		LLD 3
	STL 3		IND 0
	LDC 2		STL 5
	STL 4		LLD 3
	LDC a _{swap}		LLD 4
	STL 5		IND 0
	LLD 5		STO
	MST		LLD 4
	LLA 3		LLD 5
	LLA 4		STO
	CUP 2		RET 1
	HLT		



左辺値として評価

Small Cプログラム `*x = e;`

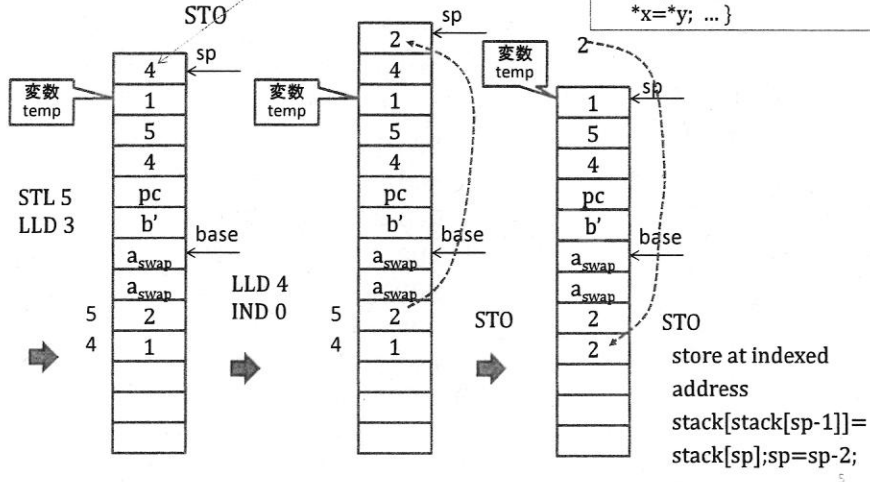
SC-コード命令 LLD 3 (xの値をロード)

eのコード LLD 4 IND 0

STO

Small Cプログラム

```
void swap (int *x, int *y){
  int temp;
  temp=*x;
  *x=*y; ... }
```



関数ポインタ、ポインタ変数 続き

Small Cプログラム

`**...*a=e;`

n個

SC-コードプログラム

aの値をロードする命令

IND 0

.

.

.

IND 0

eのコード

STO

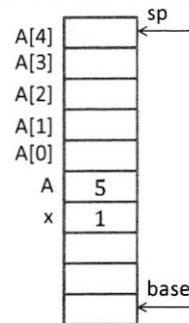
n-1個

配列とポインタ

Small Cプログラム

```
int main(void) {
  int x, A[5];
  x=1;
  A[x+2]=6;
  x=A[x+2]-7;}

```



SC-コードプログラム

```
ENT 9      STO
LLA 5      LLD 4
STL 4      LLD 3
LDC 1      LDC 2
STL 3      BOP +
LLD 4      BOP +
LLD 3      IND 0
LDC 2      LDC 7
BOP +      BOP -
BOP +      STL 4
LDC 6      HLT

```

左辺値として評価

例

例

Small Cプログラム

```
A[e]=e';
```

```
A[x+2]=6;
```

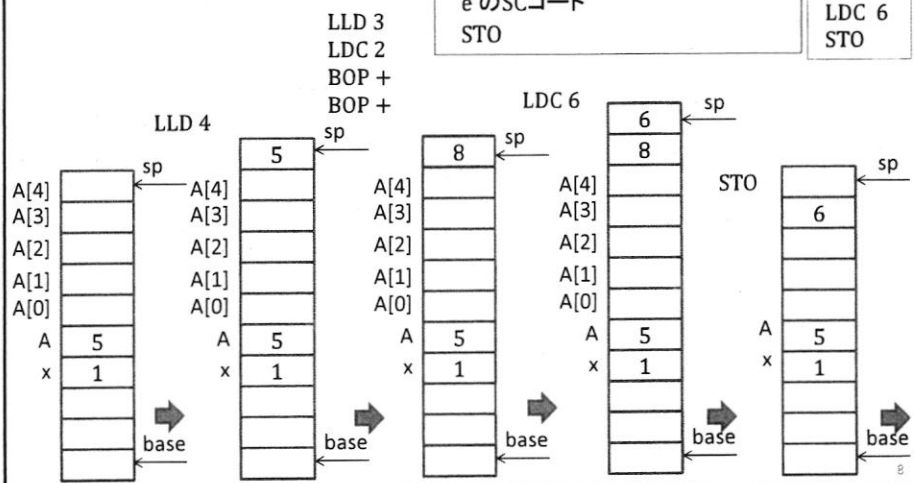
SC-コードプログラム

```
LLD &A-base (&Aの相対番地)
eのSCコード
BOP +
e'のSCコード
STO

```

```
LLD 4
LLD 3
LDC 2
BOP +
BOP +
LDC 6
STO

```



右辺値として評価

Small Cプログラム
A[e]

SC-コードプログラム
LLD &A-base (&Aの相対番地)
eのSCコード
BOP +
IND 0

問題

演習【13】 テキスト例7.5の配列要素に対応するSC-コードにおいて、

$x = A[x+2] - 7;$ の左辺の項 $A[x+2]$ に対応するSC-コードを抜き出せ。

式と文

代入演算子を含む式文

- (1) $e \text{ op } e'$ ただし op は代入演算子
 - 例) `int x, y; x = y + 2;`
- (2) $\text{op } e$ ただし op は、`&`、`++`、`--`
 - 例) `int x, *y; ++x; y = &x;`
- (3) $e \text{ op}$ ただし op は、`++`、`--`
 - 例) `int x; x--;`

ここで、式 e の左辺値の計算が必要とされる場合

- (a) $e = \tilde{e}$; 代入を示す。 $e =$ のこと
- (b) $\&e$

(a) $e = \tilde{e}$; の場合

(a) の場合に左辺値が必要な式 e は、構文図式図 7.13 より 3通り

- (i) $e = *e'$ かつ 単項式 $\Rightarrow *e'$
 - e の左辺値は、 e' の右辺値
 - e の左辺値を計算する SC-コードは、 e' の右辺値を計算する SC-コードと同じ
 - 例) `int *x, y; *(x+1) = y + 2;`
- (ii) $e = \text{ident}$
 - `ident` (すなわち、変数名) に割り当てられたスタックセルの番地が e の左辺値
 - 左辺値をプッシュする命令 `LLA` と `LGA` を用いる
 - 例) `int x, y; x = y + 2;`

(a) $e = \tilde{e}$ の場合 続き

- (iii) $e = e'[e'']$, 後置式 $\Rightarrow *e'$ かつ式 $\Rightarrow *e''$
 - $e'[e'']$ が $*(e' + e'')$ に等しい
 - e の左辺値を計算するSCコードは、 $e' + e''$ の右辺値を計算するSCコードと同じ
 - 例) `int A[5], x; x=2; A[x+3]=4;`

(b) $\&e$ の場合

- (イ) $e = \text{ident}$ のとき

e が左辺値を持つ式であるためには、非終端記号である左辺値式から生成されるものに限られる

e op=e'; のSC-コード

(イ) e=identのとき

ident op=e';

LLD &ident-base (またはLGD &ident)
e'の右辺値を計算するSCコード

BOP op

STL &ident-base (またはSTG &ident)

例) int x, y; x+=y+2;

(ロ) e≠identのとき

e op=e';

eの左辺値を計算するSCコード

DUP

IND 0

e'の右辺値を計算するSCコード

BOP op

STO

例) int *x, y;

*(x+1) += y+2;

15

e =e'; のSC-コード

(ハ) e=identのとき

ident=e';

e'の右辺値を計算するSCコード

STL &ident-base (またはSTG &ident)

例) int x, y; x=y+2;

(ニ) e≠identのとき

e=e';

eの左辺値を計算するSCコード

e'の右辺値を計算するSCコード

STO

例) int *x, y;

*(x+1) = y+2;

16

$e = e'$ が e'' の真部分式の場合

(ハ) $e = \text{ident}$ のとき

$\dots(\text{ident} = e')\dots a$

例) `int x, y, z;`

`z = (x = y + 2) + 3;`

e' の右辺値を計算するSCコード

STL &ident-base (またはSTG &ident)

LLD &ident-base (またはLGD &ident)

(二) $e \neq \text{ident}$ のとき

$\dots (e = e') \dots$

例) `int *x, y, z;`

`z = (*(x + 1) = y + 2) + 3;`

e の左辺値を計算するSCコード

e' の右辺値を計算するSCコード

STF

STF `stack[stack[sp-1]] = stack[sp]; stack[sp-1] = stack[sp]; sp--;`

副作用演算子の制限

単項演算子 $++$ 、 $--$ を含む式文のSC-コード $op \in \{++, --\}$

- $++e$; は、副作用単項式で、
 - $e += 1$; に等しい
- 式文 e ; が副作用単項式 $++e'$ を含む場合
 - $++e'$; $e[++e'$ を e' に置き換え]; に等しい
- 副作用単項式の使用制限
 - 式文中に同じ副作用単項式が現れてはいけない
 - 関数呼び出しの実引数の中で使用できない

制御文 のSC-コード

if文のSC-コード

if (e) s

eのSC-コード

FJP a

sのSC-コード

a:

if (e) s else s'

eのSC-コード

FJP a₁

sのSC-コード

UJP a₂

a₁:

s'のSC-コード

a₂:

19

制御文 のSC-コード 続き

while文のSC-コード

while (e) s

a₁:

eのSC-コード

FJP a₂

sのSC-コード

UJP a₁

a₂:

break;

UJP a;

20

式のSC-コード

$op \in \{*, /, \%, +, -, <, >, <=, >=, ==, !=, \&\&, ||\}$

① $e \ op \ e'$

eのSC-コード
e'のSC-コード
BOP op

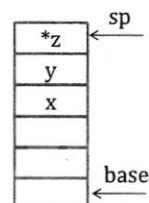
② $op \ e$

eのSC-コード
UOP op

問題

演習【14】下記のSmall C言語のプログラムから生成されるSC-コード命令を求めよ。

```
int main (void) {
    int x, y
    int *z;
    x=6; y=7;
    z=&x;
    *z=x- (1-y)/3;
}
```



関数mainが呼び出された
直後のスタック