情報実験I 第6回　課題

BP14003 秋山和哉

【課題3 考察】

[変更前]

・i1: int型の4byte分は，グローバル変数として宣言されたので，静的領域で確保される。

・i2, i3: short型の2byte分は，mainブロック内で参照可能な自動変数なので，スタック領域で確保される。

・\*i4, \*i5: char型へのポインタ型の変数は，mainブロック内で参照可能な自動変数なので，スタック領域で確保される。

・i4, i5: char型の1byte分は，mainブロック内で動的確保された変数なので，ヒープ領域で確保される。メモリマップの表記上，同じ場所を指し示しているが，ブロック毎に確保，開放して時間差が生じている為である。

・i6: short型の2byte分は，i2=-1を引数としているので，表す値は同じである。また，func1ブロック内で参照可能な自動変数なので，スタック領域で確保される。

・i7: short型の2byte分は，i3=-2を引数としているので，表す値は同じである。また，func2ブロック内で参照可能な自動変数なので，スタック領域で確保される。メモリマップの表記上，i6と一部同じ場所を指し示しているが，ブロック毎に確保，開放して時間差が生じている為である。func2ブロック最終行にて”\*i7 \*= \*i7”が行われており，実質(-2)×(-2)が計算され，元のi3へ戻っているので，表示すると”i3=4”となっている。

[変更後]

・char \*i4 🡪 static char \*i4, short i2=-1 🡪 static short i2=-1: スタック領域で確保されていたが，静的属性が付いた為に，静的領域に確保される。

・free(i4);をfree(i5);の直後に移動: 変更前，ヒープ領域にて重なっていた場所は，i4が先に動的確保，開放する前にi5が動的確保を行った為，同じ場所でなく異なった場所にプロットせざるを得なくなった。

【課題4】

[A]宣言に拠る場合

(1)アクセス手順

d / (メモリアクセス)

🡪 (short \*)(d + i \* M \* N \* sizeof(short) + j \* N \* sizeof(short) + k \* sizeof(short)) / (計算)

🡪 \*(( short \*)(d + 2iMN + 2jN + 2k)) / (メモリアクセス)

(2)計算: 1回 / メモリアクセス: 2回

(3)メモリ使用量: LMN\*sizeof(short) = 2LMN

(4)(3)に代入 🡪 3 \* 512 \*512 \* 2 / 1024 = 1536.0[KB]

(5)(3)に代入 🡪 512 \* 512 \* 3 \* 2 / 1024 = 1536.0[KB]

(6)関数への渡し方: func(d)

(7)関数での受け方: func(short (\*d) [M] [N])

[B]動的確保に拠る場合

(1)アクセス手順

d / (メモリアクセス)

🡪 d + i \* sizeof(short \*\*) / (計算)

🡪 \*(d + 8i) / (メモリアクセス)

🡪 \*(d + 8i) + j \* sizeof(short \*) / (計算)

🡪 \*(\*(d + 8i) + 8j) / (メモリアクセス)

🡪 \*(\*(d + 8i) + 8j) + k \* sizeof(short) / (計算)

🡪 \*(\*(\*(d + 8i) + 8j) + 2k) / (メモリアクセス)

(2)計算: 3回 / メモリアクセス : 4回

(3)メモリ使用量:

LMN\*sizeof(short) + LM\*sizeof(short \*) + L\*sizeof(short \*\*) + sizeof(short \*\*\*)

=2LMN + 8LM + 8L + 8

(4)(3)に代入 🡪 (2\*3\*512\*512 + 8\*3\*512 + 8\*3 + 8) / 1024 = 1536.0[KB]

(5)(3)に代入 🡪 (2 \*512 \*512 \* 3 + 8 \* 512 \*512 + 8 \* 512 + 8) / 1024 = 2562.0[KB]

(6)関数への渡し方: func(d)

(7)関数での受け方: func(short \*\*\* d)