

## ca-6. プログラムカウンタ

## (コンピュータ・アーキテクチャ演習)

URL: https://www.kkaneko.jp/cc/ca/index.html

金子邦彦









## 6-1 プログラムカウンタの振る舞い 6-2 Visual Studio でプログラムカウンタの表示 6-3 命令実行サイクル



## 6-1 プログラムカウンタの振 る舞い





# ・プログラムカウンタには、次に実行すべき命令の アドレスが入っている



## 変数 age の値が 12 以上の ときの経路















































## 6-2 Visual Studio で プログラムカウンタの表示

## プログラムカウンタの表示操作









- Visual Studio を起動しなさい
- Visual Studio で, Win32 コンソールアプリケー ション用プロジェクトを新規作成しなさい

プロジェクトの「名前」は何でもよい



# Visual Studioのエディタを使って、ソースファイルを編集しなさい Int main()

static int age, p; age = 20; if (age > 12) p = 1800; else p = 500; return O; 追加



・ビルドしなさい、ビルドのあと「1 正常終了,
 0 失敗」の表示を確認しなさい

## → 表示されなければ,プログラムのミスを自分で 確認し,修正して,ビルドをやり直す

	Ŀл	レド(B)	デバッグ(D)	チーム(M)	ツール	(T)	₹
]_	Ē.	-בעע	ションのビル	ド(В)		Ctrl+	Sh
	-	クリユーションのリビルド(N)					
-		ソリューションのクリーン(C)					
$\overline{\mathcal{V}}$		ソリューションでコード分析を実行(Y) Alt+F11			F11		
	*	ConsoleApplication4 のビルド(U)					

出力	
出力元(S): ビルド	• 🖆 🖆 🛓
- 1> ビルド開始: プロジェクト:ConsoleApplication6, - 1> stdafx.cpp - 1> ConsoleApplication6.cpp	構成:Debug Win32
1> ConsoleApplication6.vcxproj -> e:¥documents¥visual 1> ConsoleApplication6.vcxproj -> e:¥documents¥visual ====================================	studio 2015¥Projectsà <del>studio 20</del> 15¥Projectsà スキップ == <mark></mark> =======



## Visual Studioで「age = 20;」の行に、ブレー クポイントを設定しなさい





• Visual Studioで, デバッガーを起動しなさい.



- •「age = 20;」の行で,実行が中断することを確認 しなさい
- ・あとで使うので、中断したままにしておくこと





## •「age = 20;」の行で,実行が中断した状態で,逆 アセンブルを行いなさい.

デバッグ(D)     チー// M)     Nsight     ツール()       フイソドク(W)     ・     ・       ・     読行(C)     ・       ・     ボイマー町(H)       ・     デバッグの停止(E)       メベマキサン(L)     オペマキル(M)       ・     第八マクシチ(L)       ・     第八マカシテ(L)       ・     ア・反変更を追用(A)       (図)     パフォーマンスプロファイラー(F)       ・     ・       ・	) ア-キテクチャ(C) F5 Ctrl+Alt+Break Shift+F5 Alt+F10 Alt+F2 Ctrl+Alt+P	<ul> <li>テスト(S) 分析(N) ウインドク(W)</li> <li>ブレークポイント(B)</li> <li>例外の設定(X)</li> <li>出力(O)</li> <li>勤新ツールの表示(T)</li> <li>GPU スレッド(U)</li> <li>クスク(S)</li> <li>並利スタック(K)</li> <li>並利フォッチ(R)</li> <li>ウオッチ(R)</li> <li>ブォッチ(W)</li> <li>自動変数(A)</li> <li>(*)</li> </ul>	ヘルプ(H) Ctrl+Alt+B Ctrl+Alt+F2 Ctrl+Alt+F2 Ctrl+Alt+F2 Ctrl+Shift+D, K ? ?? Ctrl+Shift+D, S ? ?? Ctrl+Alt+V, A ? ?? Ctrl+Alt+V, A ? ?? Ctrl+Alt+V, A ? ??
? その他のデバッグターゲット(H) ? プロファイラー ステップイン(L)	► F11	イミディエイト(I) Python Debug Interactive	Ctrl+Alt+I ?? ?? Shift+Alt+I
( ? ステップオーバー(O) ? ステップアウト(T) ユードマップに呼び出し履歴を表示(K)	F10 Shift+F11 Ctrl+Shift+`	DOM Explorer           ライブ どジュアル ツリー(V)           マイブ ブロパティ エクスプローラー(P)	¢ (53
↔ クイックウォッチ(Q) ブレークポイントの設定/解除(G) ブレークポイントの作成(B)	Shift+F9 F9	<ul> <li>(一 呼び出し履歴(C)</li> <li>22 スレッド(H)</li> <li>(一 モジュール(O)</li> </ul>	Ctrl+Alt+C Ctrl+Alt+H Ctrl+Alt+U
<ul> <li>すべてのブレークボイントの削除(D)</li> <li>すべてのブレークボイントの無効化(N)</li> <li>IntelliTrace(I)</li> </ul>	Ctrl+Shift+F9	● プロセス(P)	Ctrl+Alt+Z
すべてのデータヒントをクリア(A) データヒントのエクフポートの			Ctrl+Alt+D 2ッサに

① 「デバッグ」→ 「ウインド ウ | → 「**逆アセンブル** |

アドレス(A): main(void) ② 表示オプション int main() { 01011650 push ebp 01011651 mov ebp,esp 01011653 sub esp,0C0h	
int main() { 01011650 push ebp 01011651 mov ebp,esp 01011653 sub esp,0C0h	
U1U11659 push ebx 01011654 push esi 01011656 push edi 01011656 lea edi,[ebp-0C0h] 01011662 mov ecx,30h 01011667 mov eax,00CCCCCCCh 01011667 rep stos dword ptr es:[edi]	
age = 20;	
<ul> <li>D101166E mov dword ptr [age (01019138h)],14h</li> <li>if (age &gt; ±12)</li> </ul>	
01011678 cmp dword ptr [age (01019138h)],0Ch 0101167F jle main+3Dh (0101168Dh) n = 1800:	
01011681 mov dword ptr [p (0101913Ch)],708h	
else 0101168B jmp main+47h (01011697h) φ = 500;	
0101168D mov dword ptr [p (0101913Ch)],1F4h	
01011697 xor eax,eax	
01011699 pop edi 01011694 pop esi	

#### ② 逆アセンブルの結果が表示される



## •「age = 20;」の行で,実行が中断した状態で,レ ジスタの中身を表示させなさい.手順は次の通り.

## EIP はプログラムカウンタ



dword ptr [age (01019138h)],14h dword ptr [age (01019138h)],0Ch main+3Dh (0101168Dh) dword ptr [p (0101913Ch)],708h

main+47h (01011697h)

dword ptr [p (0101913Ch)],1F4h

eax,eax

デバッガーを <b>起動済み</b> で,	
プログラムの実行が中断し	1
ているときに・・・	

tud	10					
8)	デバ	ッグ(D) チーム(M) N <mark>g</mark> ht ツール	(T) アーキテクチャ()	C) テス	ト(S) 分析(N) ウィン	ドウ(W) ヘルプ(H)
-	•	74フドフ(W) グラフィックフ(C) 続行(C) まがて中断(山)	F5 Ctri L Alta Brook	• •	ブレークポイント(B) 例外の設定(X) 出力(O)	Ctrl+Alt+E Ctrl+Alt+E
······································	÷.	デバッグの停止(F)	Shift+E5	227	GDU 71 w L(U)	CUITAILTI
	- × ອ ເຊິ	すべてデタッチ(L) すべて中止(M) 再起動(R)	Ctrl+Shift+F5	î M	タスク(S) 並列スタック(K) 並列ウォッチ(R)	Ctrl+Shift Ctrl+Shift
		コード変更を遮用(A) パフォーマンス プロファイラー(F) プロセスにアタッチ(P) その他のデバッグ ターゲット(H)	Alt+F10 Alt+F2 Ctrl+Alt+P	(4) (4)	ウォッチ(W) 自動変数(A) ローカル(L) イミディエイト(I)	Ctrl+Alt+ Ctrl+Alt+ Ctrl+Alt+
÷,		プロファイラー		► 100	Python Debug Interact	ive Shift+Alt+
	* ? *	ステップ イン(L) ステップ オーバー(O)	F11 F10	Ω	JavaScript コンソール(J) DOM Explorer	Ctrl+Alt+\
	÷ PT	ステッフ アット(1)	Shift+Fill	O'	ライブ ビジュアル ツリー(V)	
	~	コート(ソフルニー)の出しるに加上していい。 クイックウォッチ(〇)	Shift+F9	0.	517 70/71 19270	1-5-(P)
		ブレークポイントの設定/解除(G) ブレークポイントの設定/解除(G) ブレークポイントの作成(B) すべてのブレークポイントの削除(D)	F9 Ctrl+Shift+F9	- 1 - 2 - 3 - 3 - 3 - 3 - 3 - 3 - 3 - 3 - 3 - 3	呼び出し履歴(C) スレッド(H) モジュール(O) プロセス(P)	Ctrl+Alt+C Ctrl+Alt+I Ctrl+Alt+I Ctrl+Alt+I
	01	9 へ ( の ) レークホイントの 無効16(N)			メモリ(M)	~ · · · ·
91 91		meenmace() すべてのデータヒントをクリア(A) データヒントのエクスポート(X) データヒントのインポート(Y)			ボックシュレ (D) レジスタ(G)	Ctrl+Alt-
1		名前を付けてタンプを保存(V)				
	-03	# 1813-101N				4 AN A

12/2/3 EA = 0000000 EB = 01555800 EX = 00003400 EB = 01015550 ESC = 0101164 ED = 01117646 ED = 0101164E ED = 01117646 EP = 01117666 EP = 0111766666 EP = 011176666 EP = 01117666666 EP = 01117666666666666666666666



① 「デバッグ」 → 「ウインドウ」→「**レジスタ**」



## ステップオーバーの操作を1回ずつ行いながら, レジスタウインドウの中の EIP の変化を確認しな さい.





age = 2U; ○ 0101166E mov if (age > +12) 01011678 cmp 0101167F jle p = 1800; 01011681 mov else 0101168B jmp p = 500; 0101168D mov return 0; 01011697 xor

dword ptr [age (01019138h)],14h dword ptr [age (01019138h)],0Ch main+3Dh (0101168Dh) dword ptr [p (0101913Ch)],708h main+47h (01011697h) dword ptr [p (0101913Ch)],1F4h

eax,eax

アセンブリ言語の中で: 赤丸: ブレークポイント 黄色矢印: プログラムカウンタ

レジスタ EAX = CCCCCCCC EBX = 013C5000 ECX = 00000000 EDX = 01019590 ESI = 01011041 EDI = 0111FAB4 EIP = 0101166E ESP = 0111F9E8 EBP = 0111FAB4 EFL = 00000200

0x01019138 = 00000000





# ・最後に、プログラム実行の再開の操作を行いなさい、これで、デバッガーが終了する.







- •「age = 20;」の行を「age = 10;」に変えて,今の 手順を繰り返しなさい.
- ジャンプの様子が変わるので確認しなさい





次のプログラムでプログラムカウンタの値の変化の様子を確認しなさい.

```
int main()
{
    static int i, s;
    s = 0;
    for (i = 1; i <= 3; i++)
        s = s + (i * i);
    return 0;
}
</pre>
```



## 5-3 命令実行サイクル





## Visual C++ の プログラム



	- 0.	MOV	dword ptr [x (OCA9138h)],3
Х	= 3;	mov	dword ptr [v (00491306)] 4
V.	= 4:		uworu pti [/ (00H3130H/),4
2	,	MOV	eax,dword ptr [x (OCA9138h)]
7	= x + y:	add	eax,dword ptr [y (OCA913Ch)]
5	- ~ · /,	MOV	dword ptr [z ( <mark>UCA914Uh</mark> )],eax

いまから、この3行を図解で説明































