# Code assembleur exercice 05 : Modification d’adresse dans la pile

Afin de résoudre l’exercice 05 de la série 03, vous devez générer le code assembleur du programme C en utilisant un outil tel que **gdb** [[1]](#footnote-1) sur linux. Cela vous permettra de comprendre ce qui se passe réellement lors de l’exécution du programme C.

Ci-dessous le code assembleur correspondant au programme C.

|  |  |
| --- | --- |
| Code C | Code assembleur |
| #include <stdio.h>  void function(int a, int b, int c){  char buffer1[5];  char buffer2[10];  char \*ptr;  **ptr = buffer1 + 12;**  **\*ptr +=10;**  }  int main(){  int x;  x = 0;  function(1,2,3);  **x = 1;**  printf("%d\n",x);  } | gdb-peda$ disassemble **main**  Dump of assembler code for function **main**:  0x080484a9 <+0>: lea ecx,[esp+0x4]  0x080484ad <+4>: and esp,0xfffffff0  0x080484b0 <+7>: push DWORD PTR [ecx-0x4]  0x080484b3 <+10>: push ebp  0x080484b4 <+11>: mov ebp,esp  0x080484b6 <+13>: push ecx  0x080484b7 <+14>: sub esp,0x14  0x080484ba <+17>: mov DWORD PTR [ebp-0xc],0x0  0x080484c1 <+24>: sub esp,0x4  0x080484c4 <+27>: push 0x3  0x080484c6 <+29>: push 0x2  0x080484c8 <+31>: push 0x1  0x080484ca <+33>: call 0x804840b <function>  0x080484cf <+38>: **add esp,0x10**  0x080484d2 <+41>: **mov DWORD PTR [ebp-0xc],0x1**  0x080484d9 <+48>: sub esp,0x8  0x080484dc <+51>: push DWORD PTR [ebp-0xc]  0x080484df <+54>: push 0x8048580  0x080484e4 <+59>: call 0x8048330 <printf@plt>  0x080484e9 <+64>: add esp,0x10  0x080484ec <+67>: nop  0x080484ed <+68>: mov ecx,DWORD PTR [ebp-0x4]  0x080484f0 <+71>: leave  0x080484f1 <+72>: lea esp,[ecx-0x4]  0x080484f4 <+75>: ret  End of assembler dump.  gdb-peda$ disassemble **function**  Dump of assembler code for function function:  0x0804840b <+0>: push ebp  0x0804840c <+1>: mov ebp,esp  0x0804840e <+3>: sub esp,0x20  => **0x08048411 <+6>: lea eax,[ebp-0x9]**  **0x08048414 <+9>: add eax,0xd**  **0x08048417 <+12>: mov DWORD PTR [ebp-0x4],eax**  **0x0804841a <+15>: mov eax,DWORD PTR [ebp-0x4]**  **0x0804841d <+18>: movzx eax,BYTE PTR [eax]**  **0x08048420 <+21>: add eax,0xa**  **0x08048423 <+24>: mov edx,eax**  0x08048425 <+26>: mov eax,DWORD PTR [ebp-0x4]  0x08048428 <+29>: mov BYTE PTR [eax],dl  0x0804842a <+31>: nop  0x0804842b <+32>: leave  0x0804842c <+33>: ret  End of assembler dump. |

1. https://cs.brown.edu/courses/cs033/docs/guides/gdb.pdf [↑](#footnote-ref-1)