Programování na strojové úrovni

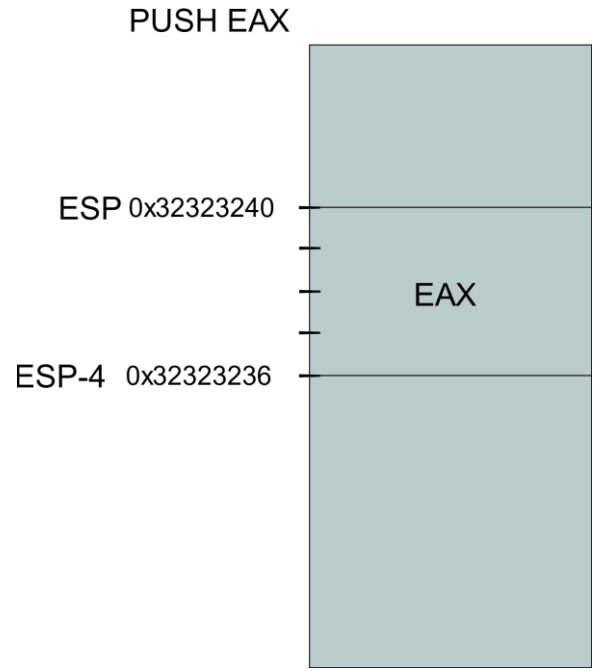
Cvičení 7: Stack overflow

Tomáš Goldmann, igoldmann@fit.vutbr.cz
www.fit.vutbr.cz/~igoldmann/courses

Rev.: 1A/2017

Instrukce pro práci se zásobníkem

- **Zásobník** - část paměti adresovatelná prostřednictvím SS:ESP. Pro práci se zásobníkem používáme instrukce PUSH a POP.
- **PUSH operand** - tato instrukce provede snížení ukazatele ESP o počet bytů určených velikosti paměťového místa nebo velikosti registru, které se na zásobník vkládají. Následně se uloží obsah paměti či registru na místo dané ukazatelem.
- **POP operand** - tato instrukce provede uložení hodnoty na místo dané operandem a následně zvýší ESP o počet bytů určených velikosti paměťového místa nebo velikosti registru, do kterého ukládáme data ze zásobníku.
- **PUSHA(D)/POPA(D)** – tyto instrukce slouží k uložení/načtení obsahů registrů na/z zásobníku. V případě, že chceme pracovat s 32 bitovými registry musíme provést explicitní vynucení použitím instrukcí PUSHAD/POPAD. Jinak pracujeme s 16bitovými registry.

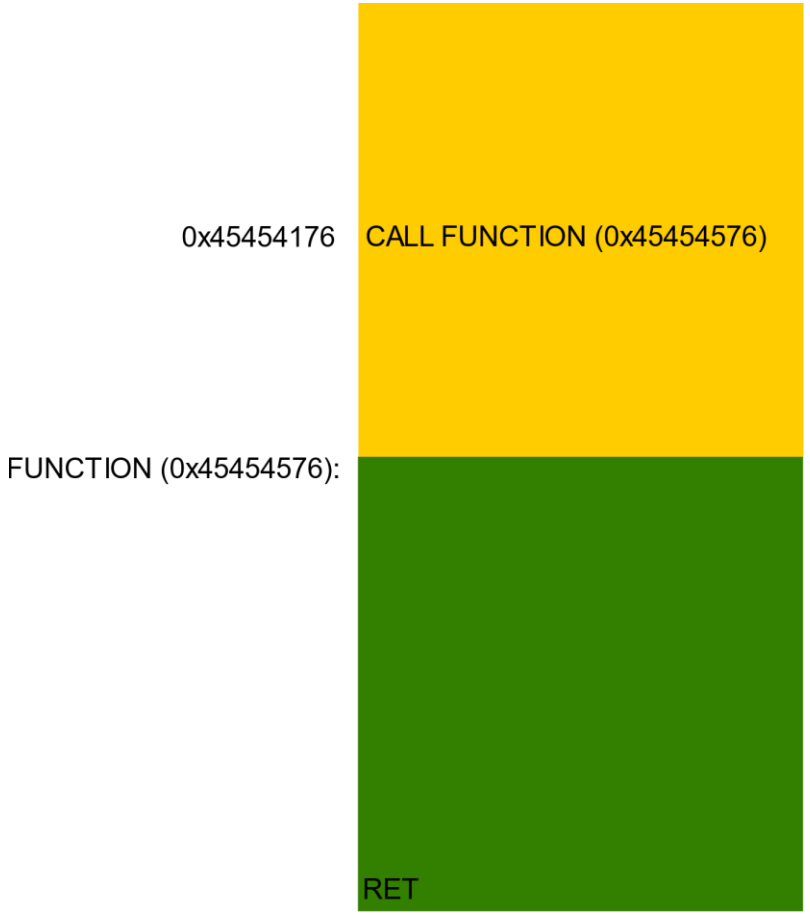


Volání funkci – bez parametrů

Instrukce pro předávání řízení – pro práci s funkcemi používáme dvě základní instrukce:

CALL – provede uložení hodnoty z registru EIP na zásobník (PUSH), následně nastaví do registru EIP hodnotu operandu.

RET – provede návrat z volané funkce. Pokud obsahuje instrukce operand, provede se následně snížení ESP o hodnotu určenou operandem.



Volání funkci – bez parametrů

Předávání parametrů volané funkcí se nejčastěji provádí nahráním parametrů na zásobník.

Aby bylo možné s těmito parametry pracovat, je zapotřebí vytvořit zásobníkový rámec.

Zásobníkový rámec – zjednodušeně – dávám nám výchozí bod

ESP → EBP

Adresování parametrů: [EBP+offset]

Adresování lokální proměnných: [EBP-offset]

Volání funkce – s paremetry (příklad)

```
Soucet:
  push ebp              ; stack-frame enter (ulozeni EBP)
  mov  ebp,esp          ; stack-frame enter (EBP ukazuje na vrchol zasobniku)

  mov  edx,[ebp+8]      ; EDX = a = 128 -- prvni argument
  mov  ecx,[ebp+12]     ; ECX = b = 2   -- druhy argument
  add  edx,ecx          ; EDX = a + b
  mov  eax,edx          ; return a + b (EAX <- EDX)

  mov  esp,ebp          ; stack-frame leave (obnoveni ESP)
  pop  ebp              ; stack-frame leave (obnoveni EBP)
  ret  8                ; stdcall odstraneni parametru ze zasobniku (2 x 32-
bitu == 8 byte)

main:

  push dword 2          ; b = 2
  push dword 128        ; a = 128
  call Soucet           ; Soucet(128,2)
ret
```

Úkoly

1. Vytvořte funkci bez parametru, která vypíše řetězec „Ahoj“
2. Vytvořte funkci se dvěma parametry, která pomocí znaku „X“ vykreslí ASCII obdélník. První parametr bude určovat šířku obdélník a druhý jeho výšku.
3. Vygenerujte pomocí volání funkce situaci „Stack overflow“
4. Vyzkoušejte si instrukce ENTER a LEAVE
5. Nahraďte všechny výskyty znaku ‚a‘ v libovolném řetězci