

variables1.c

```
int foo() {  
    int numero;  
}
```

Codigo asm generado:

```
foo:  
  
    endbr32  
    push    ebp  
    mov     ebp, esp  
    nop  
    pop     ebp  
    ret
```

- Conclusion: no se genera nada si no se inicializa.

variables2.c

```
int foo() {  
    int numero = 21;  
}
```

Codigo asm generado:

```
foo:  
  
    endbr32  
    push    ebp  
    mov     ebp, esp  
    sub     esp, 16  
    mov     DWORD PTR [ebp-4], 21  
    nop  
    leave  
    ret
```

- Conclusion: se pone en la primer posicion despues del EBP en el stack. Se reserva espacio extra (4 veces la variable en este caso)

variables3.c

```
int numero;
int foo() {
    numero = 21;
}
```

Codigo asm generado:

```
foo:
    endbr32
    push    ebp
    mov     ebp, esp
    call    __x86.get_pc_thunk.ax
    add     eax, OFFSET FLAT:_GLOBAL_OFFSET_TABLE_
    mov     eax, DWORD PTR numero@GOT[eax]
    mov     DWORD PTR [eax], 21
    nop
    pop     ebp
    ret
```

Generado con el flag -fno-pie

```
foo:
    endbr32
    push    ebp
    mov     ebp, esp
    mov     DWORD PTR numero, 21
    nop
    pop     ebp
    ret
```

- Conclusion: se guarda en la GLOBAL OFFSET TABLE.

variables4.c

```
int foo() {
    static int numero = 21;
}
```

Codigo asm generado:

```

foo:
    endbr32
    push    ebp
    mov     ebp, esp
    call    __x86.get_pc_thunk.ax
    add     eax, OFFSET FLAT:_GLOBAL_OFFSET_TABLE_
    nop
    pop     ebp
    ret
    .size   foo, .-foo
    .data
    .align 4
    .type   numero.1504, @object
    .size   numero.1504, 4
numero.1504:
    .long   21

```

- Conclusion: se guarda en la GLOBAL OFFSET TABLE. Se inicializa en la seccion .data

variables5.c

```

extern int numero;
int foo() {
    numero = 10;
}
int numero = 21;
int bar() {
    numero = 30;
}

```

Codigo asm generado:

```

foo:
    endbr32
    push    ebp
    mov     ebp, esp
    call    __x86.get_pc_thunk.ax
    add     eax, OFFSET FLAT:_GLOBAL_OFFSET_TABLE_
    mov     DWORD PTR numero@GOTOFF[eax], 10
    nop
    pop     ebp
    ret

.size     foo, .-foo
.globl    numero
.data
.align    4
.type     numero, @object
.size     numero, 4

numero:
    .long   21
    .text
    .globl  bar
    .type   bar, @function

bar:
    endbr32
    push    ebp
    mov     ebp, esp
    call    __x86.get_pc_thunk.ax
    add     eax, OFFSET FLAT:_GLOBAL_OFFSET_TABLE_
    mov     DWORD PTR numero@GOTOFF[eax], 30
    nop
    pop     ebp
    ret

```

- Conclusion: se guarda en la GLOBAL OFFSET TABLE. Se inicializa en la seccion .data. De manera global/inicial vale 21 a menos que una funcion lo cambie.

variables6.c

```

static int numero = 10;

int foo() {
    numero = 20;
}

```

Codigo asm generado:

```
.file    "variables6.c"
.intel_syntax noprefix
.text
.data
.align 4
.type    numero, @object
.size    numero, 4
numero:
    .long    10
    .text
    .globl   foo
    .type    foo, @function
foo:
    endbr32
    push    ebp
    mov     ebp, esp
    call    __x86.get_pc_thunk.ax
    add     eax, OFFSET FLAT:_GLOBAL_OFFSET_TABLE_
    mov     DWORD PTR numero@GOTOFF[eax], 20
    nop
    pop     ebp
    ret
```

- Conclusion: se guarda en la GLOBAL OFFSET TABLE. Se inicializa en la seccion .data. Se inicializa en 10 a menos que una funcion lo cambie.