

Plancha 2 Ejercicios 1, 2, 3 y 4 Representación Computacional de Datos

Arquitectura del Computador

Hedman Ulises, Pitinari Tomás y Quintero Iago

1)a)

```
.global main
main:
movb $0xFF, %al
ret
```

b)

```
.global main
main:
movb $0xFE, %al
movb $-1, %bl
addb %bl, %al
incb %bl
ret
```

a) El código de salida del programa en assembler retorna 0377, el cual está en octal, $(0377)_8 = (255)_{10} = (FF)_{16}$

```
int main() {
    return 255;
}
```

b) El código de salida del programa en assembler retorna 0375, el cual está en octal, $(0375)_8 = (253)_{10} = (FD)_{16}$

```
int main(){
    int al = 254;
    int bl = -1;

    al += bl;

    bl +=1;

    return al;
}
```

2) Para ver el código de salida de un programa podemos:

- Debugearlo con gdb y ver el registro %rax antes de que termine el programa,

```
gcc archivo.s
gdb ./a.out

(gdb) run
Starting program: /ruta/archivo/a.out
[Inferior 1 (process 45013) exited with code 0375]
```

- Ejecutar el programa en la consola

```
gcc archivo.s
./a.out
echo $?
```

3)a)

(a) 10000000 00000000 00000000 00000000 == $1 \ll 31$

```
.global main
main:
    movl $1, %eax
    shll $31, %eax
    ret
```

(b) 10000000 00000000 10000000 00000000 == $(1 \ll 31) \mid (1 \ll 15)$

```
.global main
main:
    movl $1, %eax
    shll $31, %eax
    movl $1, %ebx
    shll $15, %ebx
    xorl %ebx, %eax
    ret
```

(c) 11111111 11111111 11111111 00000000 == $-1 \& (-256)$

```
.global main
main:
    movl $-1, %eax
    movl $-256, %ebx
    andl %ebx, %eax
    ret
```

(d) 10101010 00000000 00000000 10101010 == $0xAA \& (0xAA \ll 24)$

```
.global main
main:
    movl $0xAA, %eax
    movl $0xAA, %ebx
    shll $24, %ebx
    andl %ebx, %eax
    ret
```

(e) 00000000 00000000 00000101 00000000 == $5 \ll 8$

```
.global main
main:
    movl $5, %eax
    shll $8, %eax
    ret
```

(f) 11111111 11111111 11111110 11111111 == $-1 \& (\sim(1 \ll 8))$

```
.global main
main:
    movl $-1, %eax
    movl $1, %ebx
    shll $8, %ebx
    notl %ebx
    andl %ebx, %eax
    ret
```

(g) 11111111 11111111 11111111 11111111 == $0 - 1$

```
.global main
main:
    movl $0, %eax
    movl $1, %ebx
    subl %ebx, %eax
    ret
```

(h) 00000000 00000000 00000000 00000000 == $0x80000000 + 0x80000000$

```

.global main
main:
    movl $0x80000000, %eax
    movl $0x80000000, %ebx
    addl %ebx, %eax
    ret

```

b) Si es necesario usar gdb, ya que los valores que devuelven los programas se salen del rango de las variables como las que usamos antes para visualizar por la consola.

4)

```

.global main
main:
    movl $-1,%eax # Solo para este tama~no el mov pone en 0
# la parte alta del registro.
    movl $2, %ecx
    imull %ecx

    shlq $32, %rdx
    orq %rdx, %rax

    xorq %rax,%rax
    movw $-1,%ax
    movw $2, %cx
    mulw %cx

    movl $0xFFFF0000, %edx
    orl %edx, %eax

    ret

```