

## Curs 10

- movzx d, s extinde fără semn conținutul lui s în registrul d
- movsx d, s extinde cu semn conținutul lui s în registrul d
- exemplu:

```
mov ah, 0C8h
movzx edx, ah ; edx = 000000C8h
movsx ebx, ah ; ebx = 111111C8h
```

```
movsx ax, [v] ; movsx ax, byte ptr DS:[offset v] ([v] poate fi doar byte)
00401002
```

```
movzx eax, [v] ; operand size not specified ([v] poate fi byte sau word)
```

```
movzx eax, dword [v] ; eroare de sintaxă (sizeof sursă < sizeof destination)
```

```
movzx eax, [ebx] ; operation size not specified
```

```
movsx dword [ebx], ah ; syntax error (d trebuie să fie registru)
```

- problema:

Se da următoarea secvență de instrucțiuni:

```
xor edx, edx
mov dl, 0fh
```

Scrieți o secvență de instrucțiuni care înmulțește cu 4 valoare pe 64 de biți reprezentată de combinația de regiștrii edx:eax

```
shl eax, 1
rcl edx, 1
```

```
shl eax, 1
rcl edx, 1
```

- putem folosi cu jump si regiștrii sau operanzi, nu doar etichete

```
jmp eax
jmp [v]
```

- exemplu:

```
mov al, 200
mov bl, -1
cmp al, bl
```

```
jl et1 ; -56 < -1 saltul se va efectua
```

```
ja et2 ; 200 > 255 saltul nu se va efectua
```

```
jb et3 ; 200 < 255 saltul se va efectua
```

```
jg et4 ; -56 > -1 saltul nu se va efectua
```

200 nu se poate compara cu -1 pe 8 biți, 200 nu poate fi reprezentat pe 8 biți signed

- loop face dec ecx si dupa compară ecx cu 0, făcând apoi restul comenzilor
-