CURS 0x0B

ALINIEREA MEMORIEI

```c
struct MixedData
{
    char Data1;
    short Data2;
    int Data3;
    char Data4;
};
```

```c
struct MixedData  /* After compilation in 32-bit x86 machine */
{
    char Data1; /* 1 byte */
    char Padding1[1]; /* 1 byte for the following 'short' to be aligned on a 2 byte boundary
assuming that the address where structure begins is an even number */
    short Data2; /* 2 bytes */
    int Data3;  /* 4 bytes - largest structure member */
    char Data4; /* 1 byte */
    char Padding2[3]; /* 3 bytes to make total size of the structure 12 bytes */
};
```

- **de unde vine acestă schimbare?**
- **CPU-ul citește în blocuri, și vrea ca blocurile să fie aliniate**

- **cum calculăm acest padding?**
  - notăm cu *start* adresa de start
  - notăm cu *align* alinierea pe care o vrem (o putere a lui 2)

- **padding = (align – (start mod align)) mod align**
  = (align – (start & (align – 1))) & (align – 1)
  = – start & (align – 1)

- **aliniat = start + padding**
  = start + ((align – (start mod align)) mod align)
  = (start + (align – 1)) & ~(align – 1)
  = (start + (align – 1)) & –align