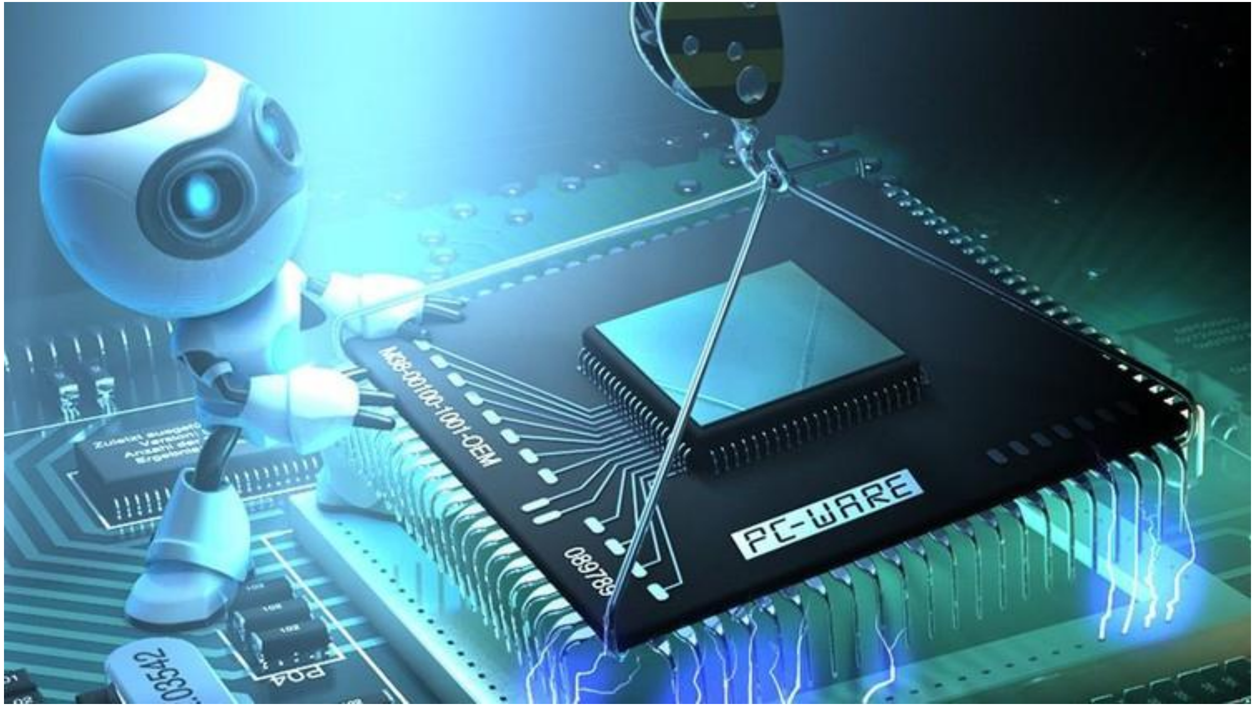
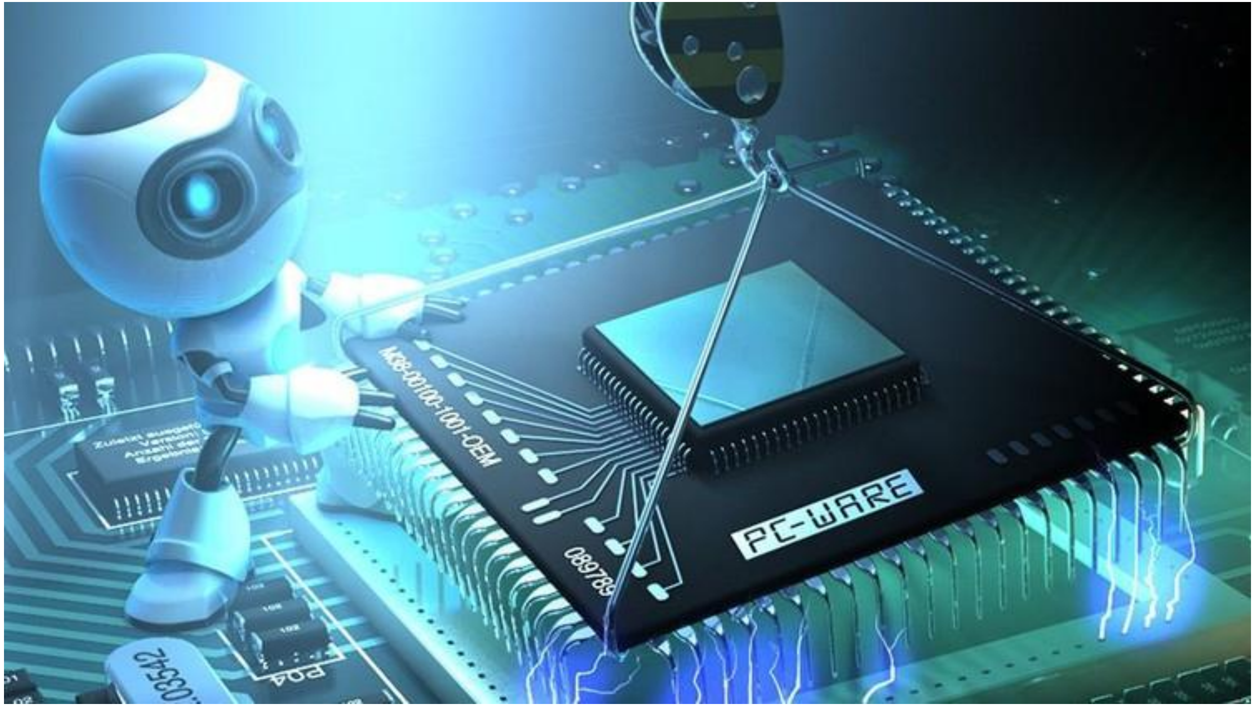
Laboratory Session 2



Practise - Problem 3



Practise - Problem 3

3. Translate the `Swap` routine to x86 assembler.

```
void Swap(S1 v[], int i, int j) {

  int tmp;
  int *aux;
  char s;

  s      = v[i].c;
  v[i].c = v[j].c;
  v[j].c = s;

  tmp    = v[i].k;
  v[i].k = v[j].k;
  v[j].k = tmp;

  aux    = v[i].m;
  v[i].m = v[j].m;
  v[j].m = aux;
}
```

Lucas Bazilio - Udemy

Practise - Problem 3

3.

Part 1/3

```
.text
        .align 4
        .globl Swap
        .type Swap,@function
Swap:
     # preparation for subroutine management
      push %ebp
      movl %esp, %ebp
      subl $12, %esp                  # we have 3 local variables (tmp,aux,s) + padding
      movl 8(%ebp), %ebx              # @v
      movl 12(%ebp), %eax             # i
      movl 16(%ebp), %edx             # j
```

Practise - Problem 3

3.

Part 2/3

```
# Swap v[i].c and v[j].c
mov (%ebx,%eax,12), %al                  # %al = v[i].c
mov (%ebx,%edx,12), %dl                  # %dl = v[j].c
mov %dl, (%ebx,%eax,12)                  # v[i].c = v[j].c
mov %al, (%ebx,%edx,12)                  # v[j].c = s

# Swap v[i].k and v[j].k
mov 4(%ebx,%eax,12), %eax                # %eax = v[i].k
mov 4(%ebx,%edx,12), %edx                # %edx = v[j].k
mov %edx, 4(%ebx,%eax,12)                # v[i].k = v[j].k
mov %eax, 4(%ebx,%edx,12)                # v[j].k = tmp
```

Previous Study - Problem 1

```
typedef struct {
   char c;
   int k;
   int *m;
} S1;
```

0	c	**1 Byte**
	padding	**3 Bytes**
4	k	**4 Bytes**
8	m	**4 Bytes**

Total size of struct S1: **12 bytes**

Practise - Problem 3

3.

Part 3/3

```
# Swap v[i].m and v[j].m
mov 8(%ebx,%eax,12), %eax          # eax = v[i].m
mov 8(%ebx,%edx,12), %edx          # edx = v[j].m
mov %edx, 8(%ebx,%eax,12)          # v[i].m = v[j].m
mov %eax, 8(%ebx,%edx,12)          # v[j].m = aux

# Function epilogue
mov %ebp, %esp
pop %ebp
ret
```