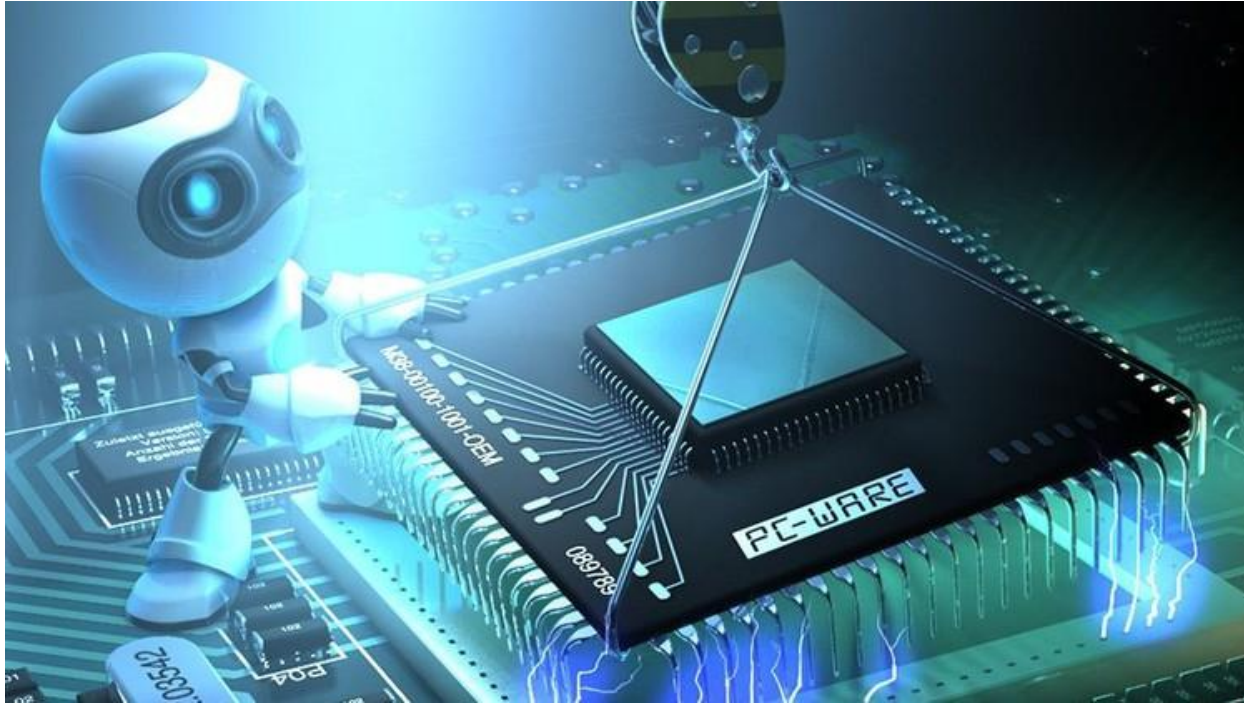


Iterative Statement (DO-WHILE)



Iterative Statement (DO-WHILE)



MODEL:

```
do
  BODY-DO
while(cond)
```

Generic translation:

```
do:    BODY-DO
       evaluate condition
       j(meets) do
end:
```

Iterative Statement (DO-WHILE)



Example:

```
int CountA(char v[]) {  
    int i, count;  
    count = 0;  
    i = 0;  
    do {  
        if (v[i] == 'a') count++;  
        i++;  
    } while (v[i] != '.');  
    return count;  
}
```

Iterative Statement (DO-WHILE)



Example:

```
int CountA(char v[]) {  
    int i, count;  
    count = 0;  
    i = 0;  
    do {  
        if (v[i] == 'a') count++;  
        i++;  
    } while (v[i] != '.');  
    return count;  
}
```

Translation:

```
CountA: pushl %ebp  
        movl %esp, %ebp  
        subl $8, %esp  
        movl $0, %eax # cont  
        movl $0, %edx # i  
do:     movl 8(%ebp), %ecx  
        cmpb $'a', (%ecx, %edx)  
        jne endif  
        incl %eax;  
endif:  incl %edx  
        cmpb $'.', (%ecx, %edx)  
        jne do  
end:    movl %ebp, %esp  
        popl %ebp  
        ret  
# @v → 8[%ebp]
```

result in % eax

Iterative Statement (DO-WHILE)



Example:

```
int CountA(char v[]) {  
  4 int i, count; 4  
    count = 0;  
    i = 0;  
    do {  
        if (v[i] == 'a') count++;  
        i++;  
    } while (v[i] != '.');  
    return count;  
}
```

Translation:

```
CountA: pushl %ebp  
        movl %esp, %ebp  
        subl $8, %esp  
        movl $0, %eax # cont  
        movl $0, %edx # i  
do:     movl 8(%ebp), %ecx  
        cmpb $'a', (%ecx, %edx)  
        jne endif  
        incl %eax;  
endif:  incl %edx  
        cmpb $'.', (%ecx, %edx)  
        jne do  
end:    movl %ebp, %esp  
        popl %ebp  
        ret  
# @v → 8[%ebp]
```

result in % eax

Iterative Statement (DO-WHILE)



Example:

`v[] = 'I have found a treasure in the sea.'`

`i = 3`

