

## ЛЕКЦИЯ 13 ПРЕМЕСТВАЕМ ОБЕКТЕН КОД

- ⌚ Проблеми на абсолютния обектен код
- ⌚ Решение на проблемите
- ⌚ Изменения в езика
- ⌚ Изменения в обектния код
- ⌚ Изменения в зареждащата програма

КА - 13

1/12

## ПРОБЛЕМИ

Привързаността на МП към определени адреси от ОП е следствие от използването на абсолютна адресация.

Тази адресация **препятства правилното изпълнение** на програма на друг адрес в ОП.

①	ORG 1000H	на адрес 1000	на адрес 2000
БРОЙ	FCB 5	1000) 05	2000) 05
СТ	ADDA БРОЙ	1001) BB 1000	2001) BB 1000
	JMP CT	1004) 7E 1001	2004) 7E 1001
②	ORG 2000H	на адрес 2000	
БРОЙ	FCB 5	2000) 05	
СТ	ADDA БРОЙ	2001) BB 2000	
	JMP CT	2004) 7E 2001	

КА - 13

2/12

## РЕШЕНИЕ

Примерът показва как може да осигурим възможност за **въвеждане** на програмата на произволно място в ОП.

Всички **адресни полета**, които **зависят от адреса на зареждане**, трябва да бъдат **отбелязани** по някакъв начин.

При **въвеждане** на друг адрес в ОП тези полета трябва да бъдат коригирани.

Един адрес на зареждане е много **удобен за изваждане** и това е **адресът 0**.

Тогава **корекциите** ще бъдат **+ нов адрес**.

КА - 13

3/12

## ИЗМЕНЕНИЯ В ЕЗИКА

В езика правят изменения, като **новият език** се нарича **Преместваем асемблер**:

- ① **Отпада** като ненужна директивата **ORG**.
- ② Символичните **имена** стават **два вида**:
  - ♣ **Абсолютни**, за които се знае съответният адрес.
  - ♣ **Преместваеми**, за които вместо съответният адрес се знае само разстоянието му до адреса на зареждане на програмата в ОП.
- ③ **Абсолютните** имена могат да участват **във всички аритметични операции**.

КА - 13

4/12

## ИЗМЕНЕНИЯ В ЕЗИКА 2

- ④ Преместваемите имена могат да участват **само** в операции **събиране** и **изваждане**:  
 $P + A \rightarrow P, P - A \rightarrow P, P - P \rightarrow A.$
- ⑤ Изразите в полето за операнди **също** са два вида: **абсолютни** и **преместваеми**.
- ⑥ Всички **АП**, създадени **чрез** **преместваеми** **изрази**, подлежат на **корекция**.
- ⑦ Такива **АП** се **маркират** и се попълват със **стойност**, която показва **разстоянието до адреса на зареждане** в ОП.

КА - 13

5/12

## ОБЕКТЕН КОД

**Преместваемият обектен код** по подобие на абсолютния също е **два вида**: образ на паметта и поредица от записи.

**Образът на паметта** при преместваемия обектен код **не е готов за изпълнение**.

**Адресът на зареждане** вече **не придружава кода**, а се **посочва при въвеждането** в ОП.

Към текста на програмата се добавя **списък на участъците**, подлежащи на **коригиране**.

**Корекцията не се посочва.**

КА - 13

6/12

## ЗАРЕЖДАНЕ В ОП

- ① Получава се **адрес на зареждане (АЗ)**.
- ② **Текстът** на програмата **се прочита** в ОП от този начален адрес АЗ.
- ③ Четат се **броят** и **списъкът** на **корекциите**.
- ④ За всяка **корекция** от списъка:
  - ⌘ към посочения адрес се **добавя АЗ**;
  - ⌘ **чете** се записаното на **получения адрес**;
  - ⌘ към прочетеното са **добавя АЗ**;
  - ⌘ получената **сума се връща** на мястото си.
- ⑤ **Към стартовия адрес** също **се добавя АЗ** за да започне изпълнение на програмата.

КА - 13

7/12

## ПОРЕДИЦА ОТ ЗАПИСИ

**Промените** в този вариант са **две**:

⌘ **адресите** във всички записи са **относителни**: **разстояние до адреса на зареждане** в ОП;

⌘ **въвежда се нов тип запис** – **корекция**.

З	0000	n	име	{		Σ	Заглавен
---	------	---	-----	---	--	---	----------

Т	адрес	n	текст	{	0,1	Σ	Текстов
---	-------	---	-------	---	-----	---	---------

С	адрес	0		Σ	Стартов
---	-------	---	--	---	---------

К	адрес	0		Σ	Корекция I
---	-------	---	--	---	------------

К	0000	n	отм.	{	Т	Σ	Корекция II
---	------	---	------	---	---	---	-------------

КА - 13

8/12

## ЗАРЕЖДАНЕ В ОП ①

При **първи вариант** на коригиращите записи въвеждането в ОП протича **по следния начин**:

- ① **получава се** адрес на зареждане (АЗ);
- ② **отново се пропускат** записите до **намиране на заглавен запис** с име на желаната програма;
- ③ при прочитане на **текстов запис** неговите ***n* байта се преместват на** посочения **адрес + АЗ**;
- ④ при прочитане на **коригиращ запис** става **четене от** посочения в него **адрес+АЗ, добавяне на АЗ** към прочетеното и **връщане** на сумата;
- ⑤ при прочитане на **стартов запис (КНФ)** започва **изпълнение от** посочения в него **адрес + АЗ**.

КА - 13

9/12

## ЗАРЕЖДАНЕ В ОП ②

При **втори вариант** на коригиращите записи **корекциите се извършват в буфера на ЗП**:

- ① **получава се** адрес на зареждане (АЗ);
- ② **намира се заглавен запис** с името на желаната за въвеждане в ОП програма;
- ③ **текстовите записи се задържат** в буфера за четене, **като предишното** му съдържание (***n* байта**) **се прехвърля на** своя **адрес + АЗ** в ОП;
- ④ при прочитане на **коригиращ запис** в буфера се извършват ***n* броя корекции** от вида: **четене, добавяне на АЗ и връщане** на сумата;
- ⑤ при прочитане на **стартов запис (КНФ)** започва **изпълнение от** посочения в него **адрес+АЗ**.

КА - 13

10/12

## ОСОБЕНОСТИ

Някои **предимства и недостатъци** на **преместваемия обектен код** са:

- ☺ МП не е привързана към конкретни адреси от ОП.
- ☺ МП може да бъде **въведена** за изпълнение на **произволно място** в ОП.
- ☺ за **преместване** не е необходим **превод**.
- ☹ МП не е **готова** за изпълнение.
- ☹ **въвеждането** на МП в ОП е **по-бавно**.
- ☹ **зареждащата програма** е **по-сложна**.

КА - 13

11/12

**БЛАГОДАРЯ ВИ  
ЗА ВНИМАНИЕТО!**

**БЪДЕТЕ С МЕН И В  
СЛЕДВАЩАТА ЛЕКЦИЯ,  
КОЯТО ЩЕ НИ ОТВЕДЕ  
В НЕВЕРОЯТНИЯ СВЯТ НА  
МЕЖДУМОДОЛНИТЕ  
ВРЪЗКИ**