

ЛЕКЦИЯ 12

АБСОЛЮТЕН ОБЕКТЕН КОД

- ⌚ **Варианти за съхраняване на машинните програми**
- ⌚ **Зареждаща програма**
- ⌚ **Образ на паметта**
- ⌚ **Поредица от записи**
- ⌚ **Сравняване на вариантите**
- ⌚ **Предимства и недостатъци**

КА - 12

1/12

СЪХРАНЯВАНЕ НА МП

Целта на транслятора от Асемблер е да се създаде **програма на машинен език**.

Тя трябва да бъде запомнена някъде.

Машинната програма може да бъде запомнена (съхранена) на **две места**:

- 👉 в **оперативната памет** на КС;
- 👉 във **външната памет** на КС.

Съхраняването във **ВП** изисква създаване на **програма за въвеждане** на програми в ОП, която се нарича **зареждаща програма**.

Зареждащата програма е **част от ОС**.

КА - 12

2/12

ЗАПИС В ОП

Записването на създадената машинна програма **в ОП е най-просто:**

- 😊 Програмата е **готова за изпълнение** (например чрез JMP СТАРТ);
- 😊 **транслаторът е по-прост.**
- 😞 **част от ОП не** може да **се използва**, защото там работи транслаторът от Асемблер;
- 😞 **всяко изпълнение** изисква **превод.**

Запомнянето на създадената машинна програма пряко в ОП е **характерно за по-старите транслатори** от Асемблер.

КА - 12

3/12

ОБРАЗ НА ПАМЕТТА

Прекият **запис в ОП не е удобен.**

В **края на I пас** става известен **размерът на програмата** в ОП и това дава възможност да моделираме ОП във ВП.

При **II пас вместо** запис в **ОП** се извършва записване в нейния **модел във ВП.**

Така във ВП получаваме точен **образ на ОП.**

Към този **образ** от 0 и 1 **се добавят:**

- 👉 **Размер** на програмата;
- ✌️ **Адрес на зареждане** на програмата в ОП;
- 👉 **Стартов адрес** на програмата.

КА - 12

4/12

ЗАРЕЖДАНЕ В ОП

При образ на паметта **зареждащата програма е проста**: трябва да се прочетат **размерът и адресът на зареждане** и след това да се извърши **една единствена операция за четене** от ВП в ОП.

След завършване на **операцията за вход** чрез **безусловен преход към** прочетения **стартов адрес** може да започне **изпълнение на въведената в ОП програма**.

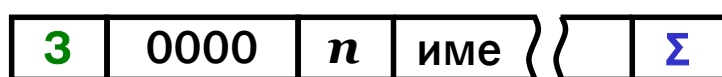
ВП може да има **ограничения за размера на физическия запис**, т. е. и на програмите.

КА - 12

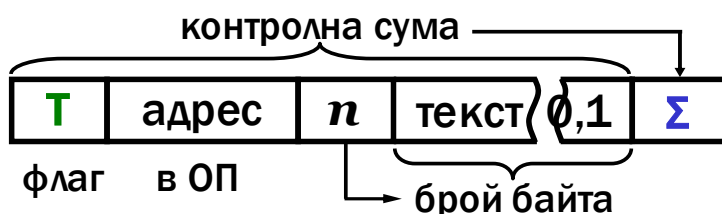
5/12

ПОРЕДИЦА ЗАПИСИ

Ограниченият **размер на физическия запис във ВП** може да бъде преодолян като **МП** се раздели на **няколко текстови (0 и 1) записа**:



Заглавен запис



Текстов запис



Стартов запис

стартов

КА - 12

6/12

ЗАРЕЖДАНЕ В ОП

При **поредица от записи зареждащата програма е малко по-сложна.**

Тя трябва да **чете записите докато намери заглавен запис с желаното име на програма.**

След това се четат **текстовите записи до откриване на стартов запис** (той е и за **КНФ**).

При прочитане на поредния **текстов запис** се определя **колко байта и на кой адрес** в ОП ще трябва да бъдат прехвърлени.

Изпълнението на програмата **започва при** прочитане на **стартов запис** чрез безусловен **преход към адреса**, посочен в този запис.

КА - 12

7/12

ЗАБЕЛЕЖКИ

При **всеки прочетен запис** трябва да **се проверява контролната сума.**

Това усложнява зареждащата програма,
но гарантира сигурност.

Припокриването на записи в ОП не се следи
за да не се усложнява зареждането. При
стартирането в ОП се намира текстът от
последния прочетен запис за даден адрес.

Тази особеност на зареждащата програма
се използва от еднопасовите Асемблери.

КА - 12

8/12

СРАВНЕНИЕ

ОБРАЗ НА ОП

- 😊 бързо въвеждане.
- 😊 по-проста зареждаща програма.
- 😞 запазените с RM участъци присъстват ненужно в образа.
- 😞 съдържанието на запазената с RM ОП не е случайно при стартиране.

ПОРЕДИЦА ЗАПИСИ

- 😊 не се извеждат записи за RM.
- 😊 съдържанието на работните полета в ОП е случайно.
- 😊 по-голяма гаранция.
- 😞 въвеждането в ОП е по-бавно.
- 😞 по-сложна зареждаща програма.

КА - 12

9/12

АБСОЛЮТЕН КОД

Чрез директива **ORG** местоположението на програмата в ОП се посочва **явно**.

Така за **всяко** символично **име** се определя неговия **точен адрес в ОП** и АП на **МИ** могат да бъдат **генерирани правилно** при превода.

Тази точност и **яснота** на съответствието **име ↔ адрес** определя и названието **АБСОЛЮТЕН** асемблер, обектен код и зареждаща програма.

Съхраняването на програмите в **абсолютен обектен код** има предимства и недостатъци.

КА - 12

10/12

ОСОБЕНОСТИ

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

- 😊 **МП** е напълно **готова** за изпълнение.
- 😊 **въвеждането** на **МП** в **ОП** е **бързо**.
- 😞 при някои видове адресация **МП** се **привързва към конкретни адреси** от **ОП**.
- 😞 **МП** не може да бъде **въведена** за изпълнение **на произволно място** в **ОП**.
- 😞 за **преместване** на ново място трябва **нов превод** с промяна на директива **ORG**.

КА - 12

11/12

**БЛАГОДАРЯ ВИ
ЗА ВНИМАНИЕТО!**

**БЪДЕТЕ С МЕН И В
СЛЕДВАЩАТА ЛЕКЦИЯ,
КОЯТО ЩЕ НИ ОТВЕДЕ
В НЕВЕРОЯТНИЯ СВЯТ НА
ПРЕМЕСТВАЕМИЯ
ОБЕКТЕН КОД**