ЛЕКЦИЯ 7
ПОДПРОГРАМИ И ПАРАМЕТРИ

- Определения
- Видове параметри в ЕПВР
- Видове параметри в МЕ
- Съглашения за предаване на параметри
- Рекурсивни подпрограми
- Съпрограми

Последна редакция 10 Септември 2005 г.

ПОДПРОГРАМИ

Подпрограмите (ППГ) в значителна степен **определят структурата** на една програма **при всеки език** без оглед на равнището му.

Подпрограма е последователност от МИ, която е **определена и се съхранява на едно място** в ОП, но може да бъде **извикана за изпълнение от** една или **няколко точки**.

Основни предимства на подпрограмите:

1. **обемът** на програмата **се намалява**;
2. **отделните** частни **задачи** се определят и обработват **от ППГ с ясно и точно** описани **взаимни връзки** с другата част на програмата.

КА-07 2/24

ПАРАМЕТРИ

Преимуществата на ППГ се проявяват **най-ярко при** използване на **параметри**.

Параметърът е **фиктивна променлива**, която се използва **при описание** на дадена ППГ.

Той осигурява **заделяне на място в ОП** като **при всяко използване** на ППГ става негово **отъждествяване с конкретна величина**.

Фиктивните променливи в дефиницията на ППГ се наричат **формални параметри**.

Величините, които се използват **при конкретното извикване** на дадена ППГ, се наричат **фактически параметри**.

КА-07 3/24

ТЕРМИНИ НА ЕПВР

Повечето **ЕПВР** вместо подпрограма използват термина **процедура**.

Параметрите на процедурите **биват**:

❶ **Входни**, чрез които процедурата **получава** начални **данни**.

❷ **Изходни**, чрез които процедурата **връща** изчислените от нея **резултати**.

Функциите са особен род **процедури**, **с един допълнителен изходен параметър**, който се отъждествява с **тяхното име**.

КА-07 4/24

ПАРАМЕТРИ В ЕПВР

Освен като входни и изходни **параметрите в ЕПВР** се различават и **по механизма на свързване** на фактическите с формалните.

В Паскал (и много други езици) **има**:

1. **параметър-стойност**: в процедурата се получава стойност с **неизвестен произход**.
2. **параметър–променлива**: в процедурата е известен произходът на стойността.
3. **параметър–процедура** и функция.
4. в **Алгол–60** има и **параметър по име**.
5. във **Фортран** има само **стойност–резултат**.

КА-07 5/24

МАШИНЕН ЕЗИК

Тъй като ППГ се използват твърде често всички ЦП имат специални МИ за извикване на ППГ: CALL, JMS, JSR, BAL и др.

В зависимост от начина за съхраняване на ПБ възвратът от ППГ се реализира чрез безусловен преход с косвена регистрова или косвена абсолютна адресация (JMP).

При съхраняване на ПБ в стек ЦП има специална инструкция за възврат: RET, RTS.

На равнище машинен език има само два вида параметри: стойности и адреси.

КА-07 6/24

СЪОТВЕТСТВИЕ

Между параметрите на **ЕПВР и МЕ има** следното **съответствие**:

параметър в ЕПВР	**съответствие в МЕ**
стойност	**стойност** на данни **с неизвестен адрес**
променлива	**адрес** на данните
процедура (ф-я)	**начален адрес** на ППГ
име (Алгол–60)	**адрес на спец. ППГ**
стойност-резултат	**адрес** на данни

КА-07 7/24

СЪГЛАШЕНИЯ

За предаване на фактическите **параметри** към дадена ППГ се използват **три системи**, известни като **„Съглашение за предаването на параметрите към подпрограми“**:

❶ **фиксирани регистри** на **ЦП**/**клетки** на ОП;

❷ **област от паметта** (**изисква** се ЦП да има базови регистри и **адресация по база**);

❸ **в стек** за параметри (така се **облекчава каскадното активиране** на редица от ППГ **и** разработването на **рекурсивни ППГ**).

КА-07 8/24

ФИКСИРАНИ РЕГИСТРИ

При тази схема параметрите се предават в предварително определени (фиксирани) регистри на ЦП, защото повечето ЦП могат да обработват само данни и адреси, които вече са записани в някой техен регистър.

☺ бърз достъп (данни в A, адреси в адресен р-р);

☺ и днес резултатът от функция е в акумулатора.

☹ ограничен брой параметри (при необходимост допълваме регистрите на ЦП с клетки на ОП);

☹ значителни проблеми при каскадни (една след друга) и, в частност, рекурсивни ППГ.

КА-07 9/24

ОБЛАСТ ОТ ПАМЕТТА

Параметрите се поместват в област от ОП, свързана с извикващата програма, а на ППГ в базов регистър се предава началният адрес на тази област.

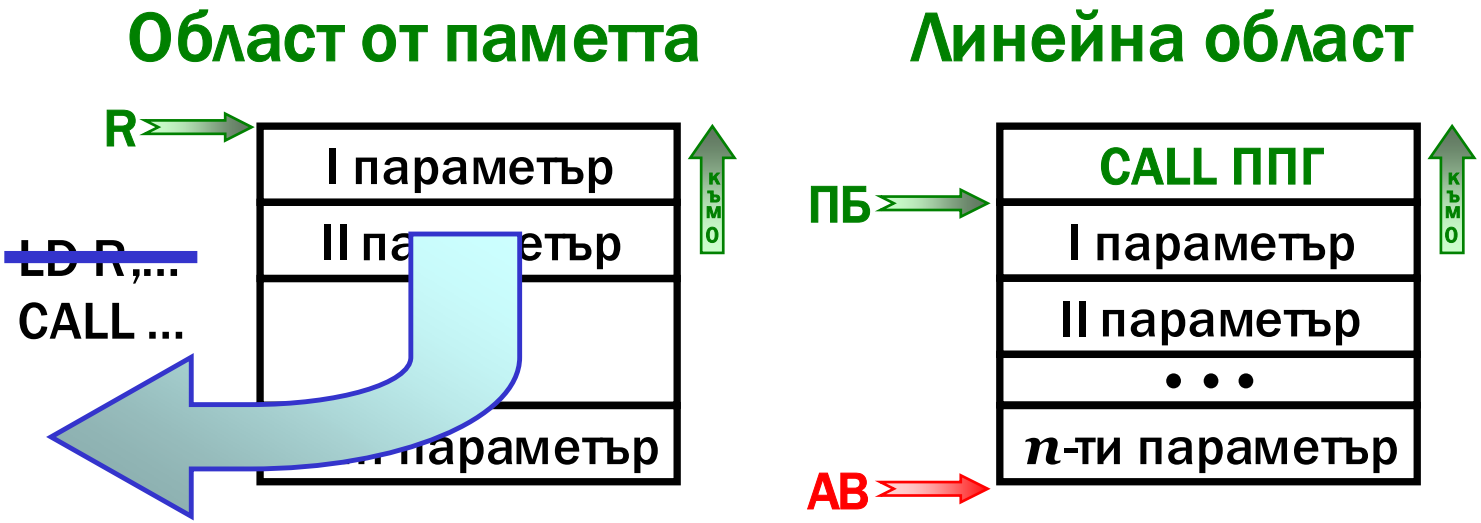
Така фактическите параметри са достъпни в подпрограмата чрез адресация по база.

Впоследствие извикващата програма може да използва същата област за предаване на параметри към друга подпрограма.

Съществува особен вариант – линейна област, която е разположена след МИ CALL, и ППГ трябва да коригира възвратния адрес.

КА-07 10/24

ПРИМЕР: ОБЛАСТ ОТ ОП



Линейна област може да **се използва само** когато **параметрите са константи**.

При променливи параметри се налага **модифициране на областта**, което **често** е **забранено**, а и **не е възможно при ПП**.

КА-07 11/24

ИЗПОЛЗВАНЕ НА СТЕК

Днес за предаване на параметри се използва стек, тъй като това е вариантът, при който каскадите и рекурсията не са проблеми. Обикновено се използва системният стек, в който се съхранява и възвратният адрес.

Стекът се използва и за динамично отделяне на място за локални променливи на ППГ.

За да се фиксира мястото, което използва ППГ, вместо системния УС се използва друг регистър, наречен Указател на Кадър (УК) от стека. Така УС може да се използва свободно за временно съхраняване на работни данни.

КА-07 12/24

ПРИМЕР: СТЕК

❶ Изпълнява се ППГ1 със свой кадър в стека.

❷ Запазва се място за изх. параметри.

❸ Записват се входните параметри.

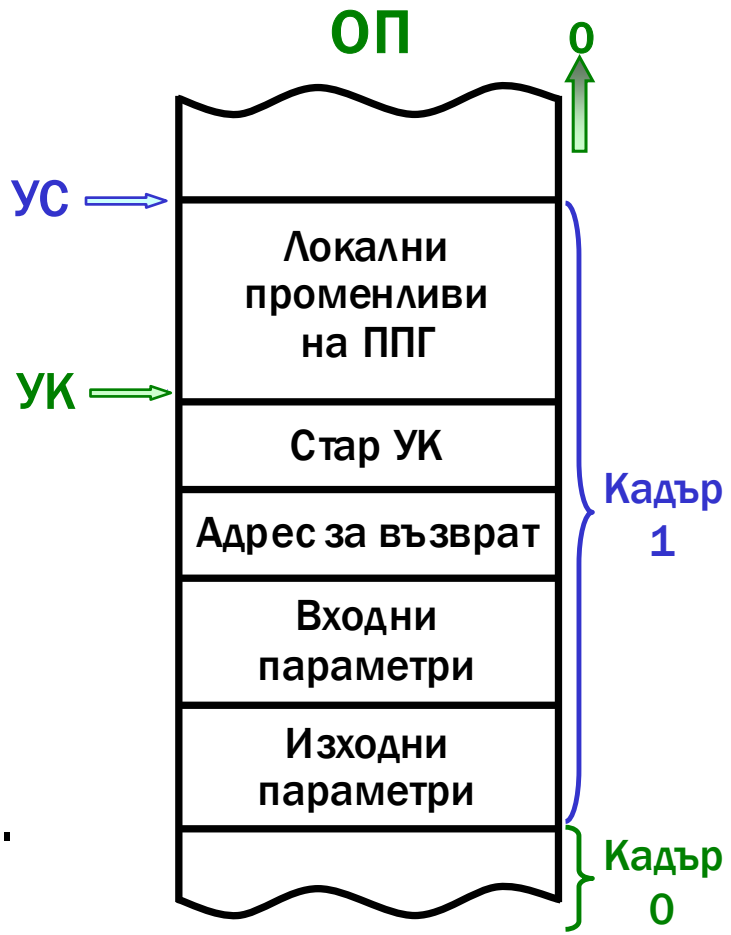
❹ Активира се ППГ2.

❺ Запазва се стария УК.

❻ Създава се нов кадър: УК := УС.

❼ Запазва се място за локални променливи.

❽ Изпълнява се ППГ2.



КА-07 13/24

ПРИМЕР: СТЕК (прод.)

❶ Подпрограмата приключва работа.

❷ Освобождава се използваният стек: УС := УК.

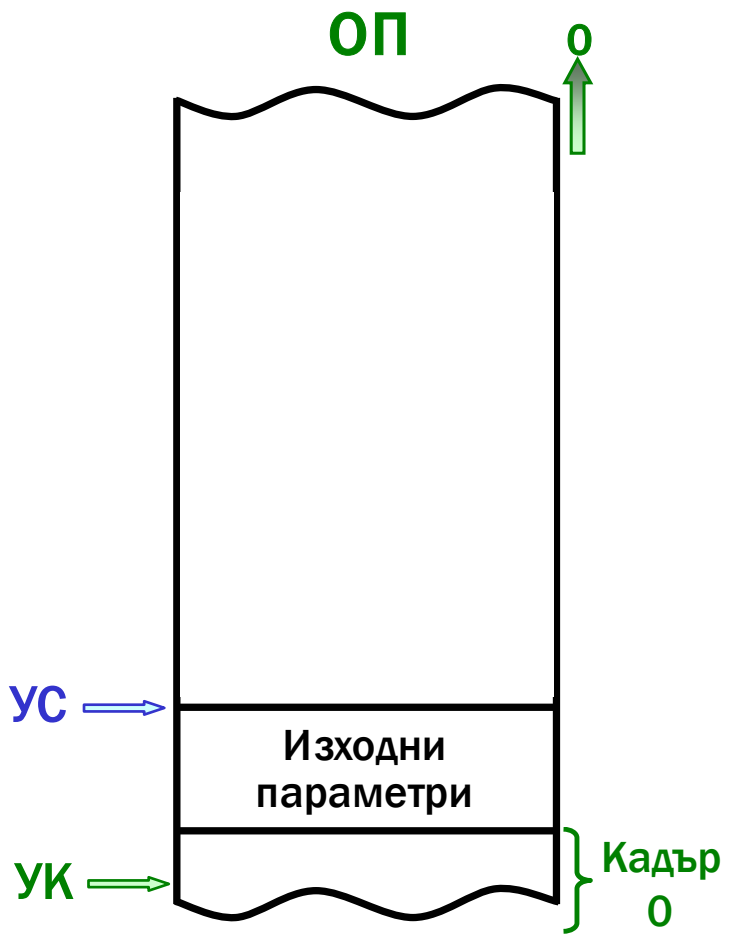
❸ Възстановява се старият УК.

❹ Получава се адресът за възврат в ППГ1.

❺ Елиминират се входните параметри.

❻ Възврат от ППГ2.

❼ Продължава ППГ1.



КА-07 14/24

ЗАБЕЛЕЖКИ

Деленето на параметрите на една ППГ **на входни и изходни**, с изключение на резултатите от функциите, в известна степен **е условно**, тъй като **отразява логиката на автора** на подпрограмата.

Затова и **практически няма ЕПВР**, който да осигурява **езикови средства за** такова **делене на обичайните параметри** при дефиниране на процедури и функции.

Изключение прави **резултатът от функция**. Той **винаги** се връща **във фиксиран регистър**.

КА-07 15/24

СХЕМИ НА ПРЕВОД ОТ ЕПВР

В чист вид описаното **съглашение**, **което предвижда разделяне** на параметрите на входни и изходни, може да **се реализира само при програмиране на МЕ**.

Компилаторите от ЕПВР прилагат при превода **две модификации**, **носещи имената на езиците**, за които са характерни и най-удобни: **Паскал и Си**.

В зависимост от свойствата на превеждания **език една от тези** две **схеми е стандартна за** работата на **съответните компилатори**.

КА-07 16/24

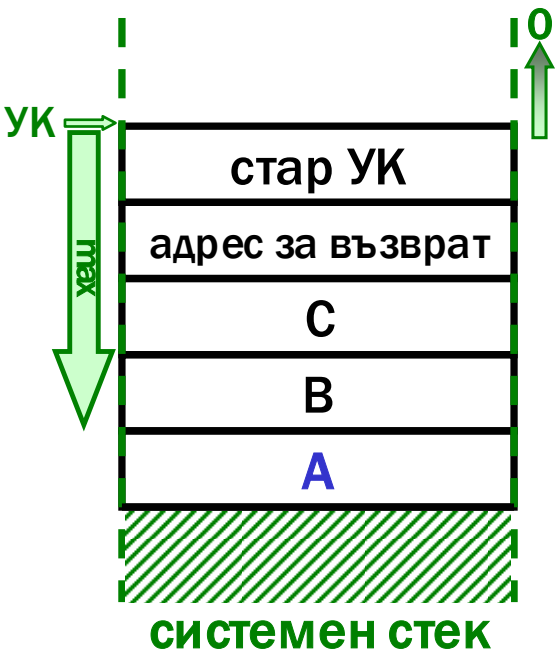
СХЕМА НА ПАСКАЛ

Езикът **Паскал** осигурява **само** ППГ с **фиксиран брой** на **параметрите**, т. е. **стекът** с параметри **ще бъде един и същ при всички активации**.

Фактическите параметри **се записват** в стека **от ляво** (**1-ви**) **на дясно** (**последен**), като **в стека най-левият** (**A**), **е най-далече спрямо УК**.

Всяка подпрограма преди възврат **почиства стека**.

```
PROCEDURE X(P1,P2, P3);
BEGIN . . . END;
. . .
X(A, B, C);
==> посока на разбора
```



КА-07 17/24

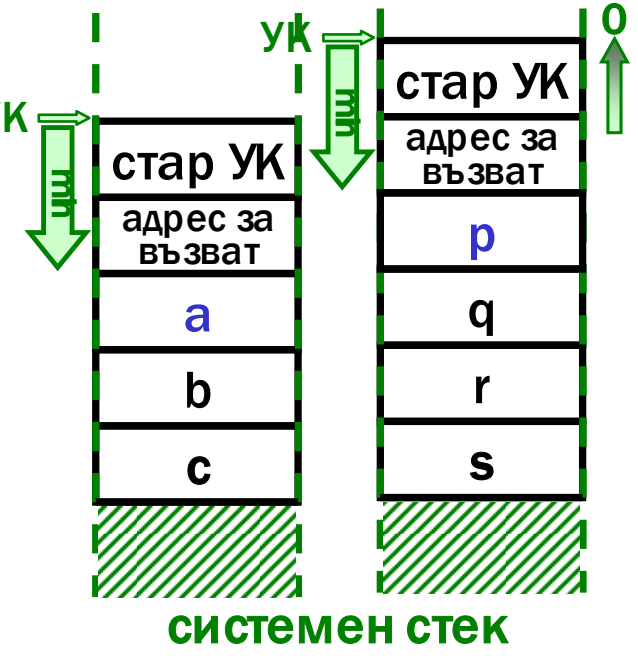
СХЕМА НА СИ

Езикът Си разрешава и ППГ с произволен брой на параметрите, т. е. стекът с параметри може да бъде различен при активациите.

Фактическите параметри се записват в стека от дясно (последен) на ляво (1-ви), като в стека най-левият (a, p), е най-близо спрямо УК.

Стекът се почиства от активиращите след възврат от всяка подпрограма.

```
void x(p1,p2,...){...}
. . .
x(a, b, c);
. . .
x(p, q, r, s);
```



КА-07 18/24

РОЛЯ НА МЕ И ЦП

Съглашението за **предаване** на параметри **чрез стек** е изключително **удобно**. Днешните ЦП осигуряват **помощ за реализиране чрез**:

❶ **адресации** авто-увеличение и авто-намаление;

❷ наличие на **системен стек** (**SS:SP**, **A15** и др.);

❸ **МИ за възврат с корекция** на УСистС (**RET** n);

❹ **МИ за съгласуване** с ППГ (**LINK** A_n,#n, **UNLK** A_n).

Като **недостатък** може да се отчете **липсата на** регистри и МИ за **контрол на сист. стек**.

ППГ могат да се пишат **и на различни езици при еднакви схеми** на двата компилатора.

КА-07 19/24

РЕКУРСИВНИ ППГ

Подпрограма, която при своето изпълнение **използва себе си**, се нарича **рекурсивна**.

Рекурсията бива **два вида**:

❶ **пряка** (проста): А използва А;

❷ **косвена**: А използва В, а В – А.

За реализиране на рекурсивни ППГ е необходимо за параметрите и локалните променливи на ППГ да **се използва стек**.

МЕ облекчава създаването на рекурсивни ППГ чрез **специфични МИ за работа със стек**, но **не осигурява МИ за контрол на стека**.

КА - 07 20/24

СЪПРОГРАМИ

Дотук разглеждахме **ППГ в контекста главна** (активираща) и **подчинена** (активирана).

При такива взаимоотношения **подчинената** трябва да завърши **изцяло своята работа до възврата в** (възобновяване на) **главната**.

Съпрограмите (копрограмите) дават възможност за заменяне на тази структура с **набор от взаимодействащи си модули**, сред които **не се определя главен модул**.

Съпрограмите трябва да **запазват** стойностите на **своите локални променливи между две активации**.

КА-07 21/24

ЗАДАЧА НА ФЛОЙД

Постановка на задачата:

❶ Изпълнете четене на редове текст, докато се срещне празен ред.

❷ Отстранете излишните интервали между думите.

❸ Отпечатайте текста по 30 знака на ред и без пренасяне на думи между два реда.

Входните и изходните потоци естествено се разделят по итеративни равнища.

Входните и изходните цикли нямат общи участъци (не се препокриват).

КА-07 22/24

РЕШЕНИЕ

Елегантното решение предвижда създаване на три съпрограми:

❶ **GetChar** чете знаци и открива празен ред;

❷ **GetWord** формира думи с отстраняване на интервалите;

❸ **PrintWord** печата редовете.

Съпрограмите се прилагат при програми, които **четат** данни, **преобразуват** ги **и** ги **извеждат**. По подобие на електронните устройства с подобни функции, такива програми се наричат **филтри**.

КА-07 23/24

БЛАГОДАРЯ ВИ ЗА ВНИМАНИЕТО!

БЪДЕТЕ С МЕН И В СЛЕДВАЩАТА ЛЕКЦИЯ, КОЯТО ЩЕ НИ ОТВЕДЕ В НЕВЕРОЯТНИЯ СВЯТ НА ПОДСИСТЕМАТА ЗА ВХОД И ИЗХОД