Элементы ЯПВУ. УКАЗАТЕЛИ.

УКАЗАТЕЛЬ – это переменная, значением которой является адрес какого-то объекта в памяти компьютера (обычно, другой переменной).

Для каждой описанной в программе переменной выделяется область памяти, когда мы объявляем в программе, например, float der, то выделяется постоянная (статическая) область памяти, которая будет занята этой переменной до конца работы программы. Указатели, в частности, позволяют организовать динамическое выделение и освобождение памяти для переменных в процессе выполнения программы.

Но указатели используют и для работы с переменными в статической памяти.

С помощью указателей например можно:

- **обрабатывать многомерные и одномерные массивы, строки, символы, структуры и массивы структур , в том числе, большого размера,**
- **динамически создавать новые переменные и структуры в процессе выполнения программы,**
- **обрабатывать связанные структуры: стеки, очереди, списки, деревья, сети,**
- **передавать функциям адреса фактических параметров,**
- **передавать функциям адреса функций в качестве параметров.**

Разберемся с организацией памяти в ПЭВМ и в программах на Pascal и C.

Адресация и распределение памяти для программ

ИНТЕРНЕТ

Для IBM совместимых ПЭВМ

Оперативная память (ОП) ПЭВМ состоит из ячеек размером в 1 байт, каждая ячейка имеет собственный уникальный номер – адрес. Память разделяется на сегменты размером 2^{16} = 65 536 байт или 64 Кб, что определяется архитектурой микропроцессоров 80x86 (16-разрядных). [32-х и 64-х-разрядная адресация поддерживается в операционных системах Windows и UNIX / Linux.]

Адрес каждого байта ОП формируется из двух слов (каждое размером по 16 бит – по 2 байта) – адреса сегмента и смещения (смещение указывает на сколько байт от начала сегмента сдвинут адрес конкретного байта).

Адрес байта формируется следующим образом: адрес сегмента смещается на 4-е двоичных разряда (бита) влево и к нему прибавляется смещение, таком образом физический адрес байта составляет 20 бит и с его помощью можно адресовать 2^{20} байт (1 Мбайт):

Указатель занимает в памяти 4 байта (32 бита) и адресует первый байт данных, например, первый байт первого элемента массива.

Указатель занимает в памяти 4 байта (32 бита) хотя собственная длина указателя – 20 бит и он помещается в 3 байта, НО обычно память выделяется словами по 2 байта (16 бит) и выравнивается на границу слова – т.е под указатель выделяется 2 слова (а не 1,5).

Адресация и распределение памяти для программ

Pascal

C / C++

Для IBM совместимых ПЭВМ и MS DOS

Для кода программы, данных и переменных, тип которых описан в программе, **компилятор** выделяет фиксированную память, не изменяющуюся в процессе выполнения программы – **статическую память**. Если невозможно заранее определить объем данных, то надо перейти к использованию **динамической памяти**. Её выделение под данные и освобождение осуществляется **программистом.** Расположение **динамической памяти** (**heap**, **кучи**) в компьютере показано на схемах ниже. Обращение к динамическим переменным осуществляется через **указатели**.

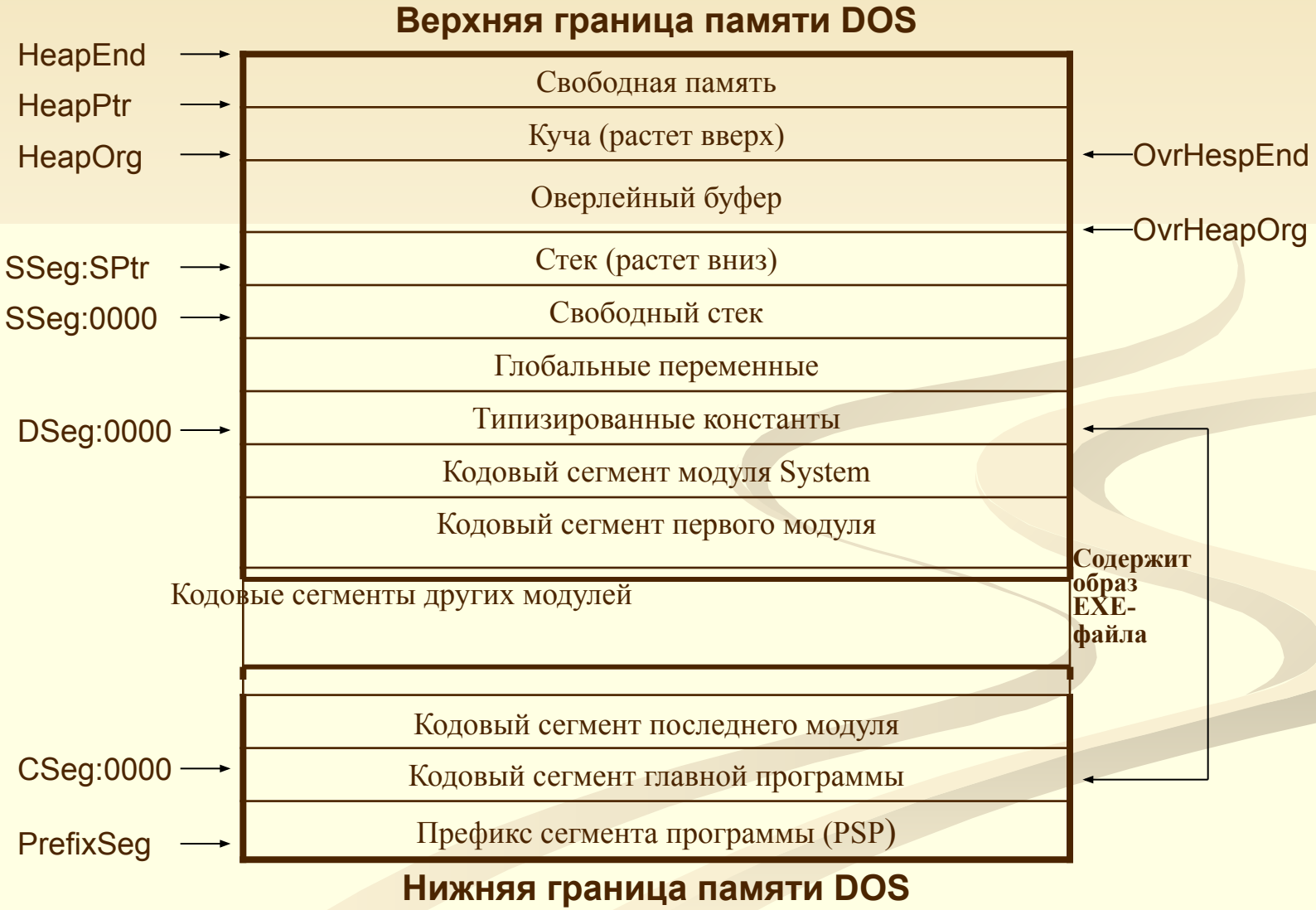
Распределение оперативной памяти ПЭВМ для программ на Pascal и C.

<table>
<tr><td>Старшие адреса</td><td colspan="2">Системная область</td></tr>
<tr><td></td><td colspan="2">Область динамической памяти (heap, куча)</td></tr>
<tr><td></td><td>Стек</td><td rowspan="3">Область программы</td></tr>
<tr><td></td><td>Данные</td></tr>
<tr><td></td><td>Код</td></tr>
<tr><td>Младшие адреса</td><td colspan="2">Системная область</td></tr>
</table>

<table>
<tr><td>Старшие адреса</td><td>Системная область</td></tr>
<tr><td></td><td>Область стека</td></tr>
<tr><td></td><td>Область динамической памяти (heap, куча)</td></tr>
<tr><td></td><td>Область глобальных переменных</td></tr>
<tr><td></td><td>Область программы</td></tr>
<tr><td>Младшие адреса</td><td>Системная область</td></tr>
</table>

Адресация и распределение памяти для программ

Карта оперативной памяти программ на Borland Pascal



Адресация и распределение памяти для программ

Замечания к карте оперативной памяти для программ на Pascal

Префикс сегмента программы (Program Segment Prefix - PSP) -это 256-ти байтовая область, создаваемая DOS при загрузке программы. **Адрес сегмента PSP хранится в переменной PrefixSeg.**

Главная программа, и каждый модуль имеют свой **кодовый сегмент. Главная программа занимает первый кодовый сегмент; кодовые сегменты, которые следуют за ним, занимают модули (в порядке, обратном тому, как они следовали в операторе uses), и последний кодовый сегмент занимает библиотека времени выполнения (модуль System). Размер одного кодового сегмента не может превышать 64К, но общий размер кода ограничен только имеющейся памятью.**

Сегмент данных (адресуемый через DSeg) содержит все глобальные переменные и затем все типизованные константы. **Регистр DS никогда не изменяется во время выполнения программы. Размер сегмента данных не может превышать 64К.**

При запуске программы регистр **сегмента стека (SSeg)** и **указатель стека (SP)** устанавливаются так, что **SS:SP указывает на первый байт после сегмента стека.** Регистр SS никогда не изменяется во время выполнения программы, а SP может передвигаться вниз пока не достигнет конца сегмента. **Размер стекового сегмента не может превышать 64К; размер по умолчанию - 16К, он может быть изменен директивой компилятора $M.**

Буфер оверлеев используется стандартным модулем Overlay для хранения оверлейного кода. Размер оверлейного буфера по умолчанию соответствует размеру наибольшего оверлея в программе; если в программе нет оверлеев, размер буфера оверлеев равен 0. Размер буфера оверлеев может быть увеличен с помощью вызова программы OvrSetBuf модуля Overlay; в этом случае размер кучи соответственно уменьшается, смещением вверх HeapOrg.

Элементы ЯПВУ. УКАЗАТЕЛИ.

Pascal

C / C++

УКАЗАТЕЛЬ – это переменная, значением которой является адрес какого-то объекта в памяти компьютера (обычно, другой переменной).

В Pascal есть два вида указателей – **типизированные** – объявляются с использованием символа ^ –

Type <тип_указателя> = ^<имя_типа_переменной>;

Type pchar = ^char; Var pc : pchar;

Можно объявить указатель прямо при описании переменной –

Var p1 : ^integer; p2 : ^real;

[В Pascal указатели, как исключение, могут ссылаться на еще не объявленный тип данных],

и **нетипизированные** – объявляются с использованием стандартного типа данных POINTER –

Var <имя_указателя> : Pointer;

Var p : pointer;

(их удобно использовать для данных, структура и тип которых изменяется в ходе работы программы).

Указатель в Pascal – переменная ссылочного типа.

Указатель в С не является отдельным типом данных, он всегда связан с каким-либо конкретным типом.

В С / С++ различают три вида указателей: **на объект**, **на функцию** и **на void**.

Объявление указателя на объект:

<Характеристики_типа_данных> * <Характеристики_указатедя>

Обычно: <тип_данных> * <имя_указателя>;

Например: int *a, *c;

Объявление указателя на void – применяется когда конкретный тип объекта, адрес которого требуется хранить, не определен:

Например: extern void *malloc(int);

Объявление указателя на функцию:

<тип> (*<имя>) (<список_типов_аргументов>);

Например: int (*fun) (double, double);

Примеры:

```
int *pi;                  // Указатель на целую переменную
const int *pci;           // Указатель на целую константу
int * const cp=&i;        // Указатель-константа на переменную
const int *const cpc=&ci; // Указатель-константа на целую константу
```

Элементы ЯПВУ. УКАЗАТЕЛИ.

Pascal

Операции с указателями

C / C++

- **Адрес переменной – @W –** возвращает адрес переменной W.
- **Разадресация (разыменование) – <имя_указателя>^ – это обращение к значению переменной, адрес которой хранится в указателе.** Указатели стандартного типа (pointer) разыменовывать нельзя.
- **Присваивание – <имя_указателя> := <ссылочное выражение>**

  Ссылочное выражение – обязательно того же типа, что и указатель, это:
  - указатель (ссылочная переменная),
  - пустая ссылка nil – нет конкретного объекта,
  - ссылочная функция (результат – ссылка, указатель).

  Например: Var p1, p2:^real; p3:^integer; pp : pointer;
  Тогда – p1:=p2; p2:=pp; - допустимо, а p2:=p3; - запрещено.
- **Сравнение – <имя_указателя_1> = [или <>] <имя_указателя_2> – сравнивать можно на равенство или неравенство.**

- **Получение адреса – &<имя_переменной> – значение адреса операнда <имя> в памяти.** Например: n = &p1;
- **Разыменование (разадресация, раскрытие ссылки) – *<имя_указателя> – значение объекта, адрес которого <имя_указателя>.** Например: z = *n;
- **Присваивание – <имя_указателя1> = <имя_указателя2> - если оба операнда имеют один тип – простое присваивание, иначе – присваивание с преобразованием типа указателя – это:**
- **Приведение типов – два вида – с указателем типа void* и без его использования.**

  void* в C разрешает неявное преобразование (в C++ – только явное, даже для void*). Преобразования всех остальных типов указателей должны быть явными, т.е. должна быть указана операция приведения типов.

  Например: если x – имеет тип int, а *p – double, то надо записывать p = (double *) &x.

  Но использовать приведение надо осторожно и обязательно к базовому типу указателя, а не объекта.

Элементы ЯПВУ. УКАЗАТЕЛИ.

Pascal

C / C++

Операции с указателями

Некоторые стандартные функции для работы с указателями:

- **addr (x) : pointer** - аналогично @,
- **seg (x) : word** – адрес сегмента для x,
- **ofs (x) : word** – смещение для x,
- **cseg : word** – значение регистра сегмента кода программы – CS,
- **dseg : word** – значение регистра сегмента данных программы – DS,
- **ptr (seg, ofs : word) : pointer** – по заданному сегменту и смещению формирует адрес типа pointer.

- **Арифметические операции (адресная арифметика)** – суммирование и вычитание – учитывают размер типа величины и применимы только к указателем одного типа:
  - **сложение (вычитание) с константой** – p1 = p1 + 12; p2 = p2 – 5;
  - **увеличение (инкремент)** – p1++; – при увеличении на 1 указатель p1 будет смещаться на величину типа данных и указывать на следующее значение Например, в типе int* – смещение на 2 байта – p1=200, p1++ = 202, p1-- = 198.
  - **уменьшение (декремент)** – p1--; – аналогично инкременту.
  - **вычитание двух указателей** – разность двух указателей – это разность их значений, деленная на размер типа в байтах. Так можно определять к-во объектов между адресами указателей.
    Например: в массиве разность указателей на 3-й и 6-й элементы равна 3.
- **Сравнение** - допустимы любые операции сравнения, обычно, они имеют смысл, когда ссылки на общий объект.

Элементы ЯПВУ. УКАЗАТЕЛИ.

Pascal

C / C++

ПРИМЕР

Вывод на печать адреса (значения указателя) в Pascal не поддерживается.

В языке C/C++ с указателями допустимы практически все операции.

```
Program Pointer;
Var
  pa : ^word;
  b: integer;
Begin
  pa^ := 100;
  b := 200;
 pa := @b;
WriteLn('pa^=',pa^,', b=',b,');
End.
```

Результат:

pa^=200, b=200

```
#include <iostream.h>
void main ()
{
int *a, b;
*a=100; b=200'
cout<<"\na="<<a<<", *a="*a<<", b="<<b;
}
```

Результат:

a=0x034c, *a=100, b=200