

Видове памет. Указатели и работа с тях

Статична, автоматична и динамична памет.
Указатели. Използване, свойства. Адресна аритметика.

Памет. Видове.

- ▶ Какво е памет.
- ▶ Йерархия на паметта в компютъра
- ▶ Видове памет за една програма
 - ▶ Регистрова
 - ▶ Статична
 - ▶ Автоматична (стекова)
 - ▶ Динамична

Указатели

- ▶ Характеристики
 - ▶ Адрес
 - ▶ Тип
- ▶ Деклариране
 - ▶ `<тип> * <име>;`
- ▶ Приложения
 - ▶ Директна работа с масиви
 - ▶ Работа с паметта
 - ▶ Подаване на аргументи на функции
 - ▶ „лек“ достъп до големи данни

Указатели. Основни операции

▶ Указатели и константи

- ▶ Константен указател `int * const p;`
 - ▶ Не може да променяме указателя, но може да променяме данните
- ▶ Указател към константа `const int * p` или `int const * p;`
 - ▶ Можем да променяме указателя, но не и данните
- ▶ Константен указател към константа `const int * const p;`
 - ▶ Не можем да променяме нито данните, нито указателя.

▶ Основни операции

- ▶ Извличане на адрес (оператор `&`)
- ▶ Извличане на стойност (оператор `*`)
- ▶ Сравняване на адреси (оператори `==` и `!=`)
- ▶ Индексиране (оператор `[]`)

Указатели. Адресна аритметика

▶ Още операции

- ▶ Сравнение ($<$, $<=$, $>$, $>=$)
- ▶ Аритметика с число ($+$, $-$, $+=$, $-=$)
- ▶ Инкрементиране ($++$, $--$)
- ▶ Разлика на указатели ($-$)
- ▶ Преобразуване на типове

Указатели и типове

- ▶ void *
- ▶ Преобразуване между типове.
- ▶ Указатели и числа.
- ▶ NULL и nullptr - разлики и приложения.

Примери

- ▶ Аргументи на функция - Swap на две числа
- ▶ Директен достъп до масив:
 - ▶ Още един път пряка селекция
 - ▶ Най-голям и най-малък елемент в една функция

Следващия пример остава за след контролното ;)

- ▶ Динамична памет - new и правила за работа с динамична памет
 - ▶ Четене на масив и създаване на нов само от четни числа