

Память

Основы информатики.

Компьютерные основы программирования

goo.gl/X7evF

На основе CMU 15-213/18-243:
Introduction to Computer Systems

goo.gl/TDDVV

Лекция 9, 13 апреля, 2015

Лектор:

Дмитрий Подлесных, кафедра информатики 608 КПМ



Память

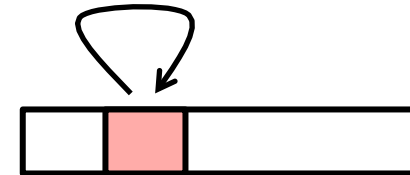
- Технологии
- Тенденции
- Локальность обращения

Локальность

- **Принцип локальности :** Программы чаще обращаются к данным и командам, чьи адреса близкие к тем которые недавно использовались

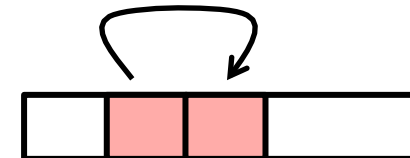
- **Временная локальность:**

- Недавно использованные ячейки вероятно вскоре будут использованы вновь



- **Пространственная локальность:**

- К ячейкам с ближайшими адресами обращения произойдут в ближайшее время



Пример локальности

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

■ Обращение к данным

- Последовательная выборка элементов массива
- Обращение к `sum` в каждой итерации

Пространственная

Временная

■ Обращение к командам

- Последовательная выборка команд
- Повторение команд цикла

Пространственная

Временная

Качественная оценка локальности

- **Утверждение:** Способность взглянув на код получить качественную оценку его локальности – ключевой навык профессионального программиста.
- **Вопрос:** Эта функция имеет хорошую локальность в отношении массива `a`?

```
int sum_array_rows(int a[M][N])
{
    int i, j, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

Пример локальности

- **Вопрос:** Вы можете переставить циклы так, чтобы функция сканировала 3-мерный массив `a` также эффективно как одномерный? С хорошей пространственной локальностью.

```
int sum_array_3d(int a[M][N][N])
{
    int i, j, k, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            for (k = 0; k < N; k++)
                sum += a[k][i][j];

    return sum;
}
```

Иерархия хранения данных

- **Некоторые фундаментальные и стабильные свойства аппаратуры и ПО:**
 - Более быстрые технологии хранения удельно дороже, менее ёмкие, и более энергоёмкие (нагрев!).
 - Пропасть «процессор-память» растёт.
 - Хороший код демонстрирует хорошую локальность.
- **Эти фундаментальные свойства замечательно дополняют друг друга**
- **Это подсказывает подход к организации подсистемы памяти известный как иерархия хранения данных.**