

Теория и практика многопоточного программирования

Семинар 1

Неганов Алексей

Московский физико-технический институт (национальный исследовательский университет)
Кафедра теоретической и прикладной информатики

Москва 2020

- Основы теории ОС
- Основы архитектуры компьютера
- Языки С и С++

- Основные концепции параллельного программирования с разделяемой памятью
- Базовые сведения об архитектуре многопроцессорных компьютеров
- Декомпозиция задач
- Синхронизация
- Разбор распространённых проблем
- Разбор работы с потоками и процессами ОС
- Ввод-вывод в многопоточной системе
- Реализация важнейших примитивов и алгоритмов на языках C и C++
- Исследование поведения параллельных программ под нагрузкой

Дополнительно:

- Демонстрация некоторых решений на языке Go
- OpenMP
- Проблемы параллельного исполнения транзакций БД
- ...

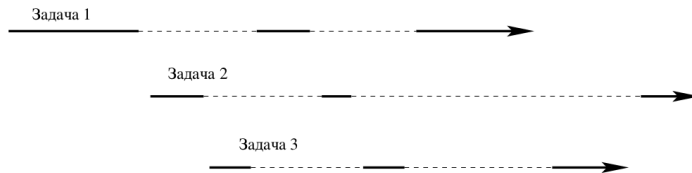


Рисунок 1 – Одновременное выполнение задач на одном процессоре

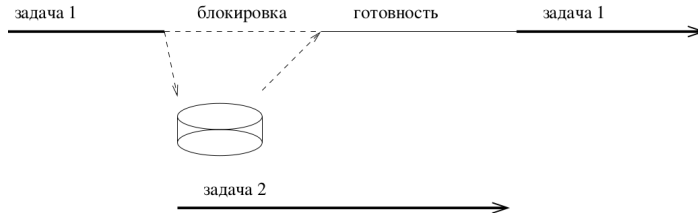


Рисунок 2 – Пакетный режим

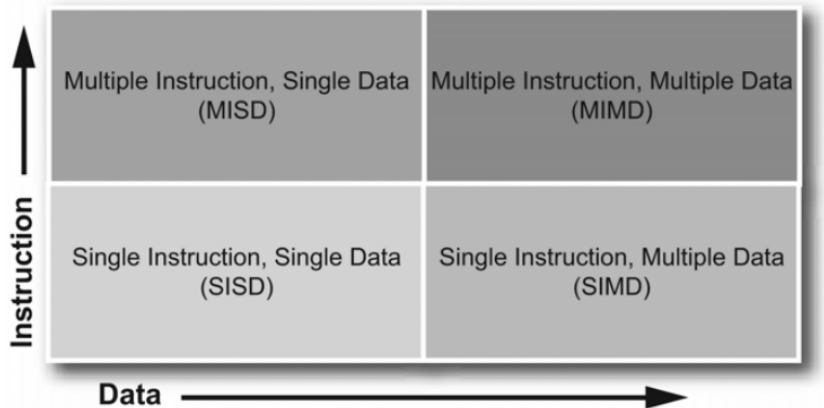


Рисунок 3 – Таксономия Флинна

Чему может оказаться равен x?

```
int x = 17;  
  
process foo {  
    x++;  
}  
  
process bar {  
    x++;  
}  
  
process baz {  
    x++;  
}
```

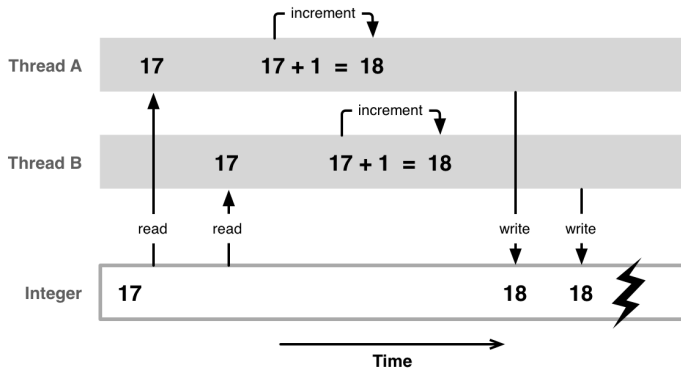


Рисунок 4 – Иллюстрация состояния гонки для оператора инкремента

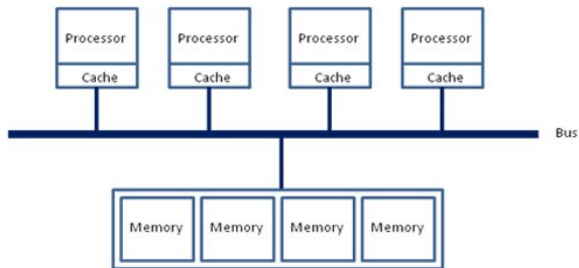


Рисунок 5 – Соединение типа SMP

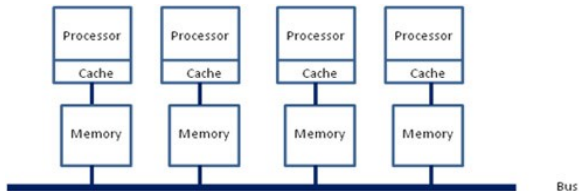


Рисунок 6 – Соединение типа NUMA

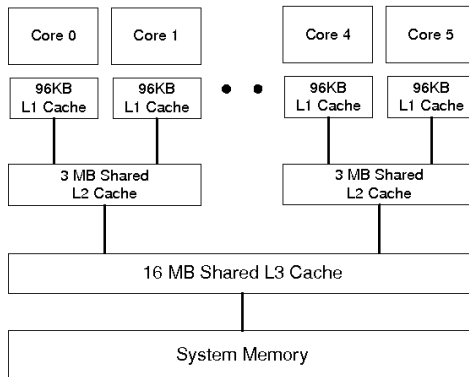


Рисунок 7 – Иерархия памяти для многоядерной SMP-системы

Может ли выполняться `assert()`?

```
int a = 0;
int b = 0;

void foo(void) {
    a = 1;
    b = 1;
}

void bar(void) {
    while(b == 0);
    assert(a == 1);
}
```

Может ли выполняться `assert()`?

```
int a = 0;
int b = 0;

void foo(void) {
    a = 1;
    smp_mb();
    b = 1;
}

void bar(void) {
    while(b == 0);
    assert(a == 1);
}
```

Чем плох этот код?

```
struct foo {  
    int x;  
    int y;  
};  
  
static struct foo f;  
  
int sum(void)  
{  
    int s = 0;  
    for (int i = 0; i < 1000000; ++i)  
        s += f.x;  
    return s;  
}  
  
void inc(void)  
{  
    for (int i = 0; i < 1000000; ++i)  
        ++f.y;  
}
```