

Title

Курс «Параллельное программирование»

Евгений Юлюгин
yulyugin@gmail.com

11 марта 2014 г.

- 1 Обзор
- 2 Классификация архитектур вычислительных систем
- 3 Состояние гонки
- 4 Синхронизация
- 5 Конец

На прошлой лекции

- Основы MPI.

Классификация Флинна

| | Single data | Multiple data |
|----------------------|-------------|---------------|
| Single instruction | SISD | SIMD |
| Multiple instruction | MISD | MIMD |

Определение

Состояние гонки (*англ.* Race condition) — ошибка в многопоточной программе, при которой работа приложения зависит от того, в каком порядке выполняются части кода.

Свое название получила от похожей ошибки проектирования электронных схем (Гонки сигналов).

Состояние гонки — ошибка проявляющаяся в случайный момент времени.

Пример

```
int N = 10;  
int x = 0;
```

```
// thread 0  
for (i = 0; i < N; ++i) {  
    x *= 2;  
}
```

```
// thread 1  
for (i = 0; i < N; ++i) {  
    x += 2;  
}
```

```
printf("%d\n", x);
```

Deadlock

TODO Write me

Livelock

TODO Write me

Алгоритм Деккера

```
bool flag[2] = {false, false};  
bool turn = false; // or true
```

```
// thread 0  
flag[0] = true;  
while (flag[1]) {  
    if (turn) {  
        flag[0] = false;  
        while (turn);  
        flag[0] = true;  
    }  
}
```

```
// critical section  
//...  
turn = true;  
flag[0] = false;  
// end of critical section  
// ...
```

```
// thread 1  
flag[1] = true;  
while (flag[0]) {  
    if (!turn) {  
        flag[1] = false;  
        while (!turn);  
        flag[1] = true;  
    }  
}
```

```
// critical section  
//...  
turn = false;  
flag[0] = false;  
// end of critical section  
// ...
```

Задания

На следующей лекции

Спасибо за внимание!

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.