

Федеральное агентство связи

Федеральное государственное образовательное бюджетное учреждение
высшего профессионального образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра ВС

Курсовая работа по дисциплине
«Моделирование» на тему:

«Имитационное моделирование вычислительной системы»

Выполнили:
магистранты ф-та ИВТ,
гр. МГ-156
Шлаузер А.И.,
Нелюбин М.А.,
Арчимаев А.С

Проверил:
проф. Родионов А.С.

Новосибирск 2016

СОДЕРЖАНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
1 Описание моделируемой системы.....	4
2 Листинг программы моделирования.....	8
3 Результаты моделирования и их анализ	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	9

ПОСТАНОВКА ЗАДАЧИ

1. Построить имитационную модель вычислительной системы, соответствующей следующему описанию:
 - вычислительная система имеет N ядер и M страниц памяти;
 - в вычислительную систему поступают задачи, требующие для выполнения определенное количество ресурсов;
 - время поступления задач, количество ресурсов определяется некоторым случайным распределением;
 - с некоторой интенсивностью в системе происходит отказ и восстановление ядер.
2. Произвести кодирование модели, провести следующие эксперименты:
 - сравнить по среднему времени пребывания задачи в системе 2 режима очереди:
 - а) FIFO;
 - б) перескок задачи через очередь по заданному времени ожидания.
 - выявить зависимость длины очереди от объема резерва.

1 Описание моделируемой системы

Формальное описание моделируемой системы можно представить в виде графа событий, изображенного на рисунке 1.

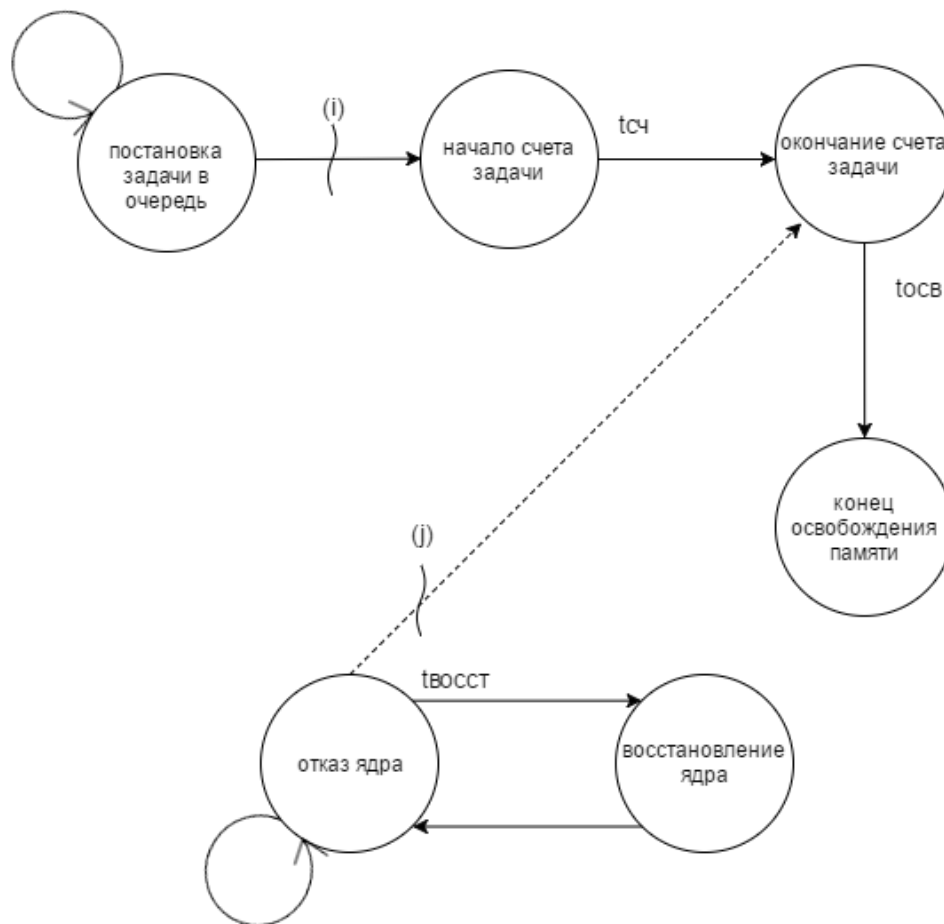


Рисунок 1 – Граф событий

Последовательную обработку событий осуществляет календарь (монитор) событий. Он извлекает из списка ближайшее по времени событие и производит действия, описанные ниже.

Постановка задачи в очередь

Созданная ранее задача добавляется в очередь. Время между приходами задач определяется экспоненциальным распределением.

Событие планируется следующим образом:

```
Generate(required_cores_num, required_memory_num, Ti)
Task = <required_cores_num, required_memory_num>
Schedule(Task, Ti, AddToQueue)
```

Ниже представлен псевдокод процедуры обработки данного события.

```
if (resources is available)
{ Schedule(Task, Ti + Twait, BeginCompute) }
```

Начало счета задачи

Событие наступает при условии i – наличии свободных ресурсов, при этом задача удаляется из очереди. Через время $t_{сч}$ планируется событие окончания счета.

Ниже представлен псевдокод процедуры обработки данного события.

```
cores_num -= required_cores_num
memory_num -= required_memory_num
Schedule(Task, Ti + Twait + Twork, EndCompute)
```

Окончание счета задачи

При наступлении данного события через время $t_{осв}$ планируется новое событие – конец освобождения памяти.

Ниже представлен псевдокод процедуры обработки данного события.

```
cores_num += required_cores_num
Schedule(Task, Ti + Twait + Twork + Trelease, FreeMemory)
```

Конец освобождения памяти

При наступлении данного события вычисляется время пребывания задачи в системе.

Ниже представлен псевдокод процедуры обработки данного события.

```
memory_num += required_memory_num
```

Отказ ядра

Отказ ядра возникает со случайным распределением. При этом ядро удаляется из системы. Если ядро было задействовано задачей, задача «убивается» (условие j). При отказе ядра через время $t_{\text{восст}}$ планируется событие восстановления ядра.

Событие планируется следующим образом:

```
Generate(crashed_core_id, Ti)
Core = < crashed_core_id>
Schedule(Core, Ti, Crash)
```

Ниже представлен псевдокод процедуры обработки данного события.

```
working_cores_count -=1
crashed_cores_count +=1
if (core is not busy) { Cancel(Task, EndCompute) }
Shedule(Core, T+recovery, Recovery)
```

Восстановление ядра

Ядро возвращается в систему.

Ниже представлен псевдокод процедуры обработки данного события.

```
working_cores_count +=1
broken_cores_count -=1
```

Также возможен другой вариант построения модели, в котором задача, находящаяся в очереди, может совершить переход в начало, не дожидаясь, когда отработают остальные. На рисунке 2 изображен граф событий для модели с «перескакиванием». Условием для «перескока» является истечение времени $t_{\text{ожид}}$.

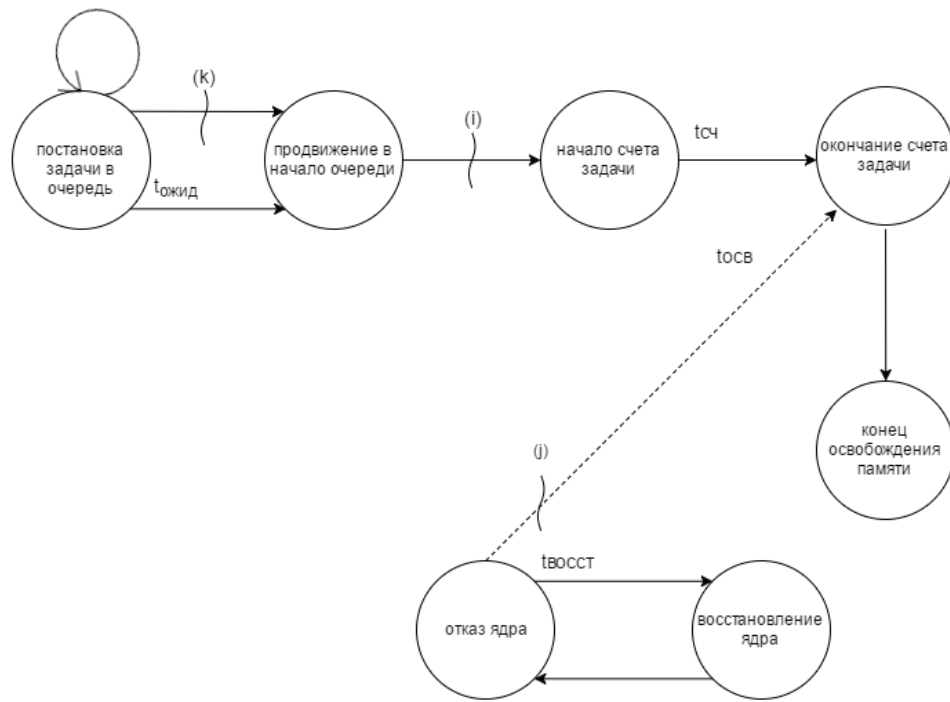


Рисунок 2 – Граф событий для модели с «перескакиванием».

- 2 Листинг программы моделирования
- 3 Результаты моделирования и их анализ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ