Федеральное агентство связи

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра ВС

Курсовая работа по предмету

«Моделирование» на тему:

Разработка сетевого приложения:

«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](#_Toc468648473)

[1 Описание моделируемой системы 4](#_Toc468648474)

[2 Листинг программы моделирования 5](#_Toc468648475)

[3 Результаты моделирования и их анализ 5](#_Toc468648476)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 6](#_Toc468648477)

# 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.

1. Произвести кодирование модели, провести следующие эксперименты:

* сравнить по среднему времени пребывания задачи в системе 2 режима очереди:

1. FIFO;
2. перескок задачи через очередь по заданному времени ожидания.

* выявить зависимость длины очереди от объема резерва.

# Описание моделируемой системы

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



Рисунок – Граф событий

Последовательную обработку событий осуществляет календарь 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.

*Начало счета задачи*

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

*Окончание счета задачи*

При наступлении данного события через время планируется новое событие – конец освобождения памяти.

*Конец освобождения памяти*

При наступлении данного события вычисляется время пребывания задачи в системе.

*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 планируется событие восстановления ядра.

*Восстановление ядра*

Ядро возвращается в систему.

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 .



Рисунок – Граф событий для модели с «перескакиванием».

# Листинг программы моделирования

# Результаты моделирования и их анализ

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.
2. Александр Шаргин. Программирование сокетов в Linux. [Электронный ресурс] URL: <http://rsdn.ru/article/unix/sockets.xml> (дата обращения: 14.12.15)
3. Sockets Tutorial. [Электронный ресурс] URL: http://www.linuxhowtos.org/C\_C++/socket.htm (дата обращения: 5.11.15)
4. Andre Adrian. TCP Chat Server in C mit Threads [Электронный ресурс] URL: http://www.andreadrian.de/thread/ (дата обращения: 05.11.15)
5. Mario Konrad  Multithreading Tutorial 4: Client/Server using Sockets and Threads. [Электронный ресурс] URL: <http://www.mario-konrad.ch/wiki/doku.php?id=programming:multithreading:tutorial-04> (дата обращения: 05.11.15)