Міністерство освіти і науки України

Національний технічний університет України

«Київський Політехнічний Інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра «Системного програмування і спеціалізованих комп’ютерних систем»

Лабораторні роботи №1-4

з дисципліни «Комп’ютерні системи» на тему:

**“** Планирование заданий в многопроцессорных системах **”**

Виконали:

студенти III курсу,  
гр. КВ-41

Яковенко Максим  
Курач Віктор

Горпинич-Радуженко Іван

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ-2017

**Мета роботи**: изучение принципов распределения ресурсов в многопроцессорных системах.

**Завдання.**

В компьютерной системе 5 процессоров. Все процессоры разные по производительности и набору команд. В систему каждую миллисекунду (мс) поступают с определенной вероятностью задачи.

Распределение ресурсов в системе может быть произведено по трем схемам:

1. **FIFO (First In First Out)** – самый простой алгоритм распределения задач. Его недостаток – низкая производительность из-за того, к примеру, что две (или более) последовательно идущие задачи, которые могут быть реализованы только на одном процессоре, могут заставить простаивать все остальные процессоры.
2. **С отдельным процессором-планировщиком.** В этом случае распределением ресурсов в системе занимается отдельный процессор, который не принимает участия в вычислениях. Так как алгоритм планирования не очень сложен, то целесообразно для этих целей выделить самый «слабый» (самый низкопроизводительный) процессор. Однако, в этом случае система реально «теряет» один процессор с точки зрения производительности и части задач, ориентированных на этот процессор. Чтобы этого избежать, необходимо докупить и добавить в систему дополнительный процессор-планировщик. Но это негативно отразится на надежности системы, т.к. выход его из строя приведет к отказу системы в целом.
3. Наиболее целесообразным представляется **возложение функций планирования на самый «мощный» (самый высоко-производительный) процессор, периодически прерывающий вычисления для управления очередью.** В этом случае считать, что время для управления очередью составляет 4 мс.

## В качестве отчета по лабораторной работе следует подать **4 набора показателей работы системы**:

а) по п. 1);

б) по п. 2);

в) по п. 3), учитывая, что время работы процессора над задачами – 20 мс, а время планирования – 4 мс.

г) по п. 3), но время работы над задачами определить самостоятельно, исходя из оптимальной производительности системы.

Показатели работы системы:

- количество реализованных задач (выполненных операций) за 10с;

- количество операций, выполненных системой за 10с.

***Примечание:*** необходимо брать среднее арифметическое количества реализованных задач (операций) для пяти испытаний по каждому из вариантов а), б), в), г).

- КПД системы;

- КПД’ системы.

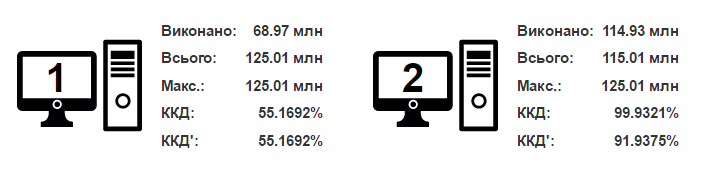
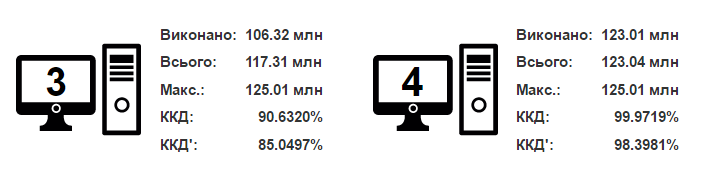
**Пояснення щодо методів розподілу задач**

***- Алгоритм FIFO (First In First Out)*** реалізований в першій частині лабораторної роботи. Кожен з процесорів, у випадку відсутності задачі для обчислення, отримує задачу, що найдовше простоюється в черзі із задач та знаходиться в стані не виконаної і не закріпленною за жодним із процесорів. В стані не можливості виконання процесором данної задачі - він чекає доки йому не потрапить з голови черги задача його рівня виконання. Під час ситуації, коли кілька задач підряд потребуватимуть процесор, котрий в певний момент часу буде зайнятий, це призведе до простою решти процесорів до тих пір, поки данний зайнятий процесор не почне виконувати задачі. В разі якщо процесор отримує певну задачу на виконання, то вона виконується лише ним, а вказівник наступної задачі переміщується на наступну задачу котра потребує обчислень.

***- Алгоритм* з окремим процесором-планувальником** розглядається другою частиною лабораторної роботи. З переглянутої черги задач для кожного з процесорів вибирається перша задача, котру він може виконати та котра не виконується іншим процесором. Дана задача оброблюється і закріплюється лише за цим процесором до її виконання. У такому випадку найслабший процесор не приймає участі в обчисленнях, проте загальний простій процесорів зменшується, що призводить до збільшення КПД.

***- Алгоритм* з накладенням функцій планування на самий потужний процесор, що періодично переривається на керування чергою** розглянуто у третій частині лабораторної роботи. Під розподіл задач виділяється певна частина робочого часу найшвидшого процесору. Через кожні 20 мс роботи процесор виконує розподіл задач за таким же алгоритмом, як це реалізовано в другій частині, тобто проходження по черзі із задач, знаходження і закріплення певної задачі на виконання тим процесором, котрий необхідний для її виконання. Найшвидший процесор виділяє задачі на виконання як іншим процесорам, так і самому собі. В даній реалізації в обчисленнях приймають участь всі процесори, що також призводить до зросту ККД.

**Результати:**

1 – алгоритм FIFO

2 – алгоритм з окремим процесором-планувальником(обирається як планувальник найслабший процесор)

3 – алгоритм з окремим процесором-планувальником(обирається найсильніший процесор, час планування – стале і рівне 4 мс, час роботи процесора – 20мс.)

4 – алгоритм з окремим процесором-планувальником(обирається найсильніший процесор, час керування чергою обирається самостійно в залежності від системи.)

**Висновки**:

1. Алгоритм FIFO очікувано виявився найгіршим по продуктивності. Так, він має показник ККД(та ККД’) близько 55.17%, що пояснюється суттєвими недоліками алгоритму: неможливість контролю розподілення задач; простій процесорів в випадку, у випадку якщо задачі, які можуть бути виконані лише на певних процесорах, будуть стояти в черзі поки обраний процесор не завершить попередню задачу, що негативно впливає на швидкодію роботи системи.
2. Алгоритм з окремим процесором-планувальником, при якому в якості планувальника виступає найменш продуктивний процесор має значно кращі показники: 99.93% по ККД та 91.94% по ККД’. Оскільки в якості планувальника виділяється найслабший процесор, то це доволі слабо впливає на швидкодію всієї системи, але поява окремого процесора, який займається плануванням виконання задач, що дає змогу зменшити час простою процесорів.
3. Алгоритм з накладенням функцій планування на самий потужний процесор, який періодично переривається на керування чергою має гірші результати ніж 2й та значно кращі результати, ніж алгоритм FIFO. Показники становлять: ККД = 90.63%, ККД’ = 85.05%. Почергове використання найбільш продуктивного процесору як планувальник та виконувач задач, пришвидшує роботу системи, в порівнянні з системою з використанням алгоритму FIFO, оскільки дає змогу більш краще розподілити задачі на процесори, але має гірші результати ніж алгоритм з використанням окремого процесора-планувальника. Очевидно, що приріст продуктивності, який дає найменш продуктивний процесор є невеликим. В випадку, якщо час розподілу часу роботи планувальника/процесора є сталим, то можливі певні простої роботи процесорів в певний момент, а найпотужніший процесор взагалі якийсь час не доступний для роботи.
4. Алгоритм з накладенням функцій планування на самий потужний процесор, який періодично переривається на керування чергою і обирає час планування в залежності від задач, що надійшли до виконання має найкращі показники ККД = 99.97%% та ККД’ = 98.40%. Він позбавлений недоліку попереднього алгоритму оскільки час роботи планувальник/виконання задач для найпродуктивнішого процесора є змінним і залежить від задач, які поступили, то алгоритми такого роду краще розпоряджаються ресурсами процесора.