МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЕ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

Кафедра «Программное обеспечение вычислительной техники и

автоматизированных систем»

ОТЧЕТ

По дисциплине: «Эвристические методы и алгоритмы»

На тему «Теория расписаний»

Выполнил студент группы ВПР32

Ли Н. С.

Проверил:

Проф. Кобак В. Г.

Ростов-на-Дону

2022

# Общие сведения

Предметом области исследования расписаний является круг задач проектирования и организационного управления в различных системах, в которых требуется найти наилучшее (оптимальное) значение выбранных критериев их функционирования с учетом имеющихся ограничений.

Программирование для многопроцессорных машинных систем связано с распараллеливанием и синхронизацией вычислений и организацией выполнения параллельных вычислительных процессов. Это выдвигает целый ряд сложных задач, среди которых весьма важными являются, расчет характеристик времени и количества операций, требующихся для выполнения параллельных программ, и построения расписаний (планов), выполнения параллельных программ на многопроцессорных и многомашинных вычислительных системах.

Модели параллельных программ и операционные характеристики процессов их выполнения служат основой для планирования параллельных вычислительных процессов, т.е. для построения расписаний указанных процессов. Расписания параллельных вычислительных процессов определяют порядок выполнения программы на вычислительной системе, включая распределение частей программы по процессам. С увеличением числа распределяемых частей программ и количества используемых процессоров сложность построения оптимальных расписаний обычно резко возрастает. Поэтому важное значение имеют простые в построении и удобные в реализации приближенные расписания параллельных вычислительных процессов, близкие к оптимальным с точки зрения времени выполнения параллельных программ.

# Постановка задачи

Имеется вычислительная система (ВС), состоящая из *N* несвязанных идентичных устройств (приборов, процессоров и т.п.)



На обслуживание в ВС поступает набор из *M* независимых параллельных заданий (работ)  известно время решения  задания  на любом из устройств. При этом каждое задание может выполняться на любом из устройств (процессоре), в каждый момент времени отдельный процессор обслуживает не более одного задания и выполнение задания не прерывается для передачи на другой процессор. Требуется найти такое распределение заданий по процессорам, при котором суммарное время выполнения заданий на каждом из процессоров было бы минимальным. Под расписанием следует понимать отображение , такое что, если , то говорят что задание , в расписании *R* назначенного на процессор . При сделанных выше допущениях, расписание можно представить разбиением множества заданий *T* на *N* непересекающихся подмножеств 

Критерий, используемый для минимизации времени завершения обслуживания заданий, является минимальным критерием и определяется в следующем виде: , где

- время завершения работы процессора .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задача.** Рассмотрим однородную матрицу N x M, где, N = 3, M = 32, где M – число заданий, N - число процессоров. | 36, 39, 36, 37, 39, 35, 35, 40, 38, 37, 40, 36, 35, 37, 35, 36, 36, 36, 37, 37, 35, 37, 38, 38, 38, 39, 40, 35, 38, 40, 38, 37 | 36, 39, 36, 37, 39, 35, 35, 40, 38, 37, 40, 36, 35, 37, 35, 36, 36, 36, 37, 37, 35, 37, 38, 38, 38, 39, 40, 35, 38, 40, 38, 37 | 36, 39, 36, 37, 39, 35, 35, 40, 38, 37, 40, 36, 35, 37, 35, 36, 36, 36, 37, 37, 35, 37, 38, 38, 38, 39, 40, 35, 38, 40, 38, 37 | 36, 39, 36, 37, 39, 35, 35, 40, 38, 37, 40, 36, 35, 37, 35, 36, 36, 36, 37, 37, 35, 37, 38, 38, 38, 39, 40, 35, 38, 40, 38, 37 |

Решим данную задачу с использованием метода алгоритма критического пути.

Для этого необходимо решить ряд задач:

1. Отсортировать список задач по убыванию приоритета
2. Для каждого элемента
   1. Выбрать минимально загруженный процессор
   2. Добавить к нему вес текущей задачи
   3. Удалить текущую задачу из очереди

Результат просчёта представлен на рис. 1:

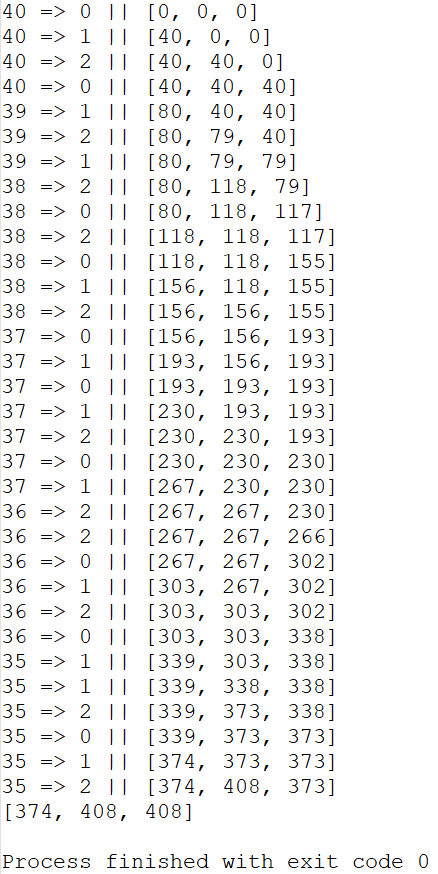


Рис. 1 – Результат работы алгоритма критического пути

Слева представлены итерируемые веса из очереди процессов. Стрелками отмечены номера минимально загруженных процессов. Справа в колонке отображено состояние загруженности процессоров на каждой итерации.

1. **Код программы**

import random

N = 3

M = random.randint(30, 35)

T = [random.randint(35, 40) for i in range(M)]

print(f"n = {N}\nm = {M}\nT = {T}")

def quicksort(nums):

if len(nums) <= 1:

return nums

else:

q = random.choice(nums)

l\_nums = [n for n in nums if n > q]

e\_nums = [q] \* nums.count(q)

b\_nums = [n for n in nums if n < q]

return quicksort(l\_nums) + e\_nums + quicksort(b\_nums)

# n - count of processors, t - list of tasks priority

def cmp(n, t):

lst = quicksort(t)

state = [0 for i in range(n)]

for el in lst:

print(f"{el} => {state.index(min(state))} || {state}")

state[state.index(min(state))] += el

return state

if \_\_name\_\_ == "\_\_main\_\_":

print(cmp(N, T))

1. **Блок-схема программы**

На рис. 2 представлена блок-схема программного средства

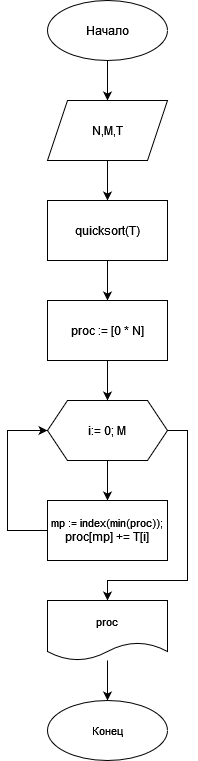


Рис. 2 Блок-схема

1. Вывод

В ходе выполнения работы было реализовано программное средство, реализующее алгоритм метода критического пути, выполнена задача в соответствии с вариантом, по методическим указаниям.

Литература:

1. Коффман Э.Г. “Теория расписания и вычислительные машины” – M.: “Наука”, 1987

2. Романовский И.В. “Алгоритмы решения экстремальных задач” – М.: “Наука”, 1977

3. Пашкеев С.Д., Минязов Р.И., Могилевский В.Д. “Машинные методы оптимизации в технике связи” – М.: “Связь”, 1976.