#### Министерство науки и высшего образования Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение науки и высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №7-8

по дисциплине

«Инструментальные средства информационных систем»

Тема: «Планирование процессов»

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Левковский Д.И.

Владимир, 2019 г.

## ЦЕЛЬ РАБОТЫ

Изучить как происходит планирование исполнения процессов в мультипрограммных вычислительных системах. Рассмотреть преимущества и недостатки различных алгоритмов планирования.

ВЫПОЛНЕНИЕ РАБОТЫ

Даны 10 процессов. Составить таблицу готовности исполнения процессов, подсчитать среднее время ожидания и среднее время выполнения, исходя из индивидуального варианта (столбец X) и при условии, что большее значение в столбце «Приоритет» соответствует меньшему приоритету. Рассмотреть вариант задания как для вытесняющего, так и для не вытесняющего алгоритмов. Составить программы по каждому варианту. (FCFI)

|  |  |  |  |
| --- | --- | --- | --- |
| Процесс | Время появления в очереди | Приоритет | Продолжительность очередного CPU burst |
| p0 | 0 | 10 | 5 |
| p1 | 2 | 9 | 4 |
| p2 | 4 | 8 | 3 |
| p3 | 6 | 7 | 2 |
| p4 | 8 | 6 | 1 |
| p5 | 8 | 5 | 6 |
| p6 | 6 | 4 | 7 |
| p7 | 4 | 3 | 8 |
| p8 | 2 | 2 | 9 |
| p9 | 0 | 1 | 10 |



Рисунок 1. Выполнение процессов первые 12 секунд.



Рисунок 2. Выполнение процессов.



Рисунок 3. Выполнение процессов.



Рисунок 4. Выполнение процессов.



Рисунок 5. Завершение выполнения процессов и подсчет времени ожидания и выполнения.

Код программы:

**class** Process:  
 number=int()  
 time = int()  
 priority = int()  
 cpuBurst = int()  
 state = **" "  
  
 def** \_\_init\_\_(self, number, time, priority, cpuBurst):  
 self.number = number  
 self.time = time  
 self.priority = priority  
 self.cpuBurst = cpuBurst  
  
ProcessList = [ Process(0, 0,10,5), Process(1, 2,9,4),  
 Process(2, 4,8,3), Process(3, 6,7,2),  
 Process(4, 8,6,1), Process(5, 8,5,6),  
 Process(6, 6,4,7), Process(7, 4,3,8),  
 Process(8, 2,2,9), Process(9, 0,1,10) ]  
  
**def** Replaning(RunningListTime):  
 minPriority = int(10)  
  
 *# находим процессы по времени и наибольший приоритет у выбраных процессов* **for** p **in** ProcessList:  
 **if** p.time <= timeNow:  
 RunningListTime[p.number] = p  
 **if** (minPriority > p.priority)&(p.state!=**""**):  
 minPriority = p.priority  
  
 *# находим процессы с наибольшим приоритетом* RunningListPriority = []  
 **for** p **in** RunningListTime:  
 **if** p.priority == minPriority:  
 RunningListPriority.append(p)  
  
 *# меняем статусы процессов* **if** (len(RunningListPriority) == 1):  
 proc = RunningListPriority[0]  
 i = RunningListTime.index(proc)  
 RunningListTime[i].state = **"и"  
 for** p **in** RunningListTime:  
 **if** (p.cpuBurst != 0) & (p.state != **"и"**):  
 p.state = **"г"  
 else**:  
 time = int(0)  
 pr=Process  
 **for** i **in** range(0,len(RunningListPriority)):  
 **if**(RunningListPriority[i].time<=time):  
 pr=RunningListPriority[i]  
 **for** p **in** RunningListTime:  
 **if** p==pr: p.state=**"и"  
 if** (p.cpuBurst != 0) & (p.state != **"и"**):  
 p.state = **"г"**quit = int(0)  
stop = 10  
timeNow = int(1)  
  
expection=int(0)  
execution=int(0)  
  
**while**(stop>0):  
 RunningListTime = [Process] \* 10  
 RunningListPriority = []  
  
 *# перепланирование* Replaning(RunningListTime)  
  
 *# вывод процессов* **for** p **in** RunningListTime:  
 **if**(p.state==**"г"**): expection+=1  
 **if**(p.state==**"г"**)|(p.state==**"и"**): execution+=1  
 print(p.state, **" "**, end=**""**)  
  
 *# проверка статусов и актуальности процессов* **for** p **in** RunningListTime:  
 **if** (p.state == **"и"**) & (p.cpuBurst != 0):  
 p.cpuBurst -= 1  
 **if** (p.state == **"и"**) & (p.cpuBurst == 0):  
 stop -= 1  
 i = ProcessList.index(p)  
 ProcessList.pop(i)  
 Replaning(RunningListTime)  
  
 print()  
 timeNow+=1  
 quit-=1  
  
print(**"время ожидания: "**, expection/10,**"\nвремя выполнения: "**, execution/10)

### 

Рисунки 6-9. Выполнение программы.

### ВЫВОД

В ходе лабораторной работы было изучено планирование исполнения процессов в мультипрограммных вычислительных системах и рассмотрены преимущества и недостатки различных алгоритмов планирования.