26 -27 <https://do2.rcokoit.ru/mod/assign/view.php?id=4246264>

23 https://do2.rcokoit.ru/mod/assign/view.php?id=4195397

22 <https://do2.rcokoit.ru/mod/assign/view.php?id=4191561>

24 <https://do2.rcokoit.ru/mod/assign/view.php?id=4197129>

28

25 <https://www.guru99.com/ru/shortest-job-first-sjf-scheduling.html>

25 https://www.guru99.com/round-robin-scheduling-example.html

# **22 Задание**

Дан непустой текст, оканчивающийся символом 'h'. Проверить, является ли текст записью числа в 16-ричной системе (без знака). Ответ (является/не является) напечатать (вывести в регистр

mov @str,r0

; проверка на число

isHex:

mov 0,r3

movb (r0)+,r1

cmp 68,r1; проверка на h

jz end2

mov r1,r2

cmp 30,r2 ;Проверка на 0

jz end2 ; проверка пройдена

sub 39,r2 ;проверка на 9

jle isHex

jmp letters

;проверка на символ

letters:

mov 66,r2 ;проверка на f

cmp r1,r2

jg end1 ; если > то это не число

mov 61,r2 ;проверка на a

cmp r1,r2

jge isHex ;если >= то символ прошёл проверку

mov 46,r2 ;проверка на F

cmp r1,r2

jg end1 ;если > то это не число

mov 41,r2 ; проверка на A

cmp r1,r2

jge isHex ;если >= то символ прошёл проверку

jmp end1 ; проверка не пройдена

end1:

mov 0,r0

system 11

stop

end2:

mov 1,r0

system 11

stop

str:

data "A123fh"

# **23 Задание**

Дана последовательность символов (отличных от точки), за которой следует точка. Определить, сбалансирована ли эта последовательность по круглым скобкам. Ответ: ДА (1) или НЕТ (0) вывести в регистр.

mov @str,r0

mov 0,r2

l:

movb (r0)+,r1

cmp 2e,r1 ;проверка на точку

jz res

cmp 28,r1 ; проверка на (

jz plus ; если символ = (

cmp 29,r1 ; проверка на )

jz minus ; если символ = )

jmp l

plus:

add 1,r2 ; счётчик +1

jmp l

minus:

sub 1,r2 ; счётчик - 1

jmp l

res:

cmp 0,r2

jz pos ;если счётчик = 0 то данные сбалансированы

mov 0,r0

system 12

stop

pos:

mov 1,r0

system 12

stop

str:

data "(())."

# **24 Задание**

Дан массив из 10 беззнаковых слов (2 байта). Инвертировать биты старших байтов всех элементов массива. Найти сумму четных элементов.

mov @array,r0

mov a,r1

mov 0,r3

zikl:

cmp 0,r1

jz finish ; если =0 проверка окончена

sub 1,r1 ; счётчик -1

mov (r0)+,r2 ;взятие символа по адресу

xor ff00, r2 ;инвертирование

and 0001, r2 ;проверка на чётность

cmp 1, r2

jz zikl

sub 2, r0

mov (r0), r2 ; взятие символа по адресу

xor ff00, r2 ;инвертирование

add r2, r3 ;сумма чётных ++

add 2, r0

jmp zikl ;проверка следующего символа

finish:

mov r3,r0

system 12

stop

array:

data ff0a

data ff0a

data ff0a

data ff0a

data ff0a

data ff0a

data ff0a

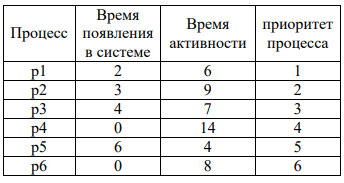
data ff0a

data ff0a

data ff0a

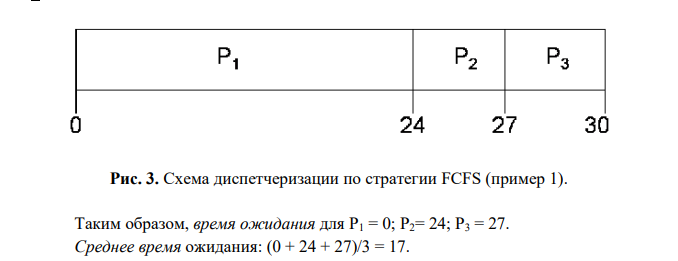
# **25 задание**

***Содержание компетентностно-ориентированного задания***



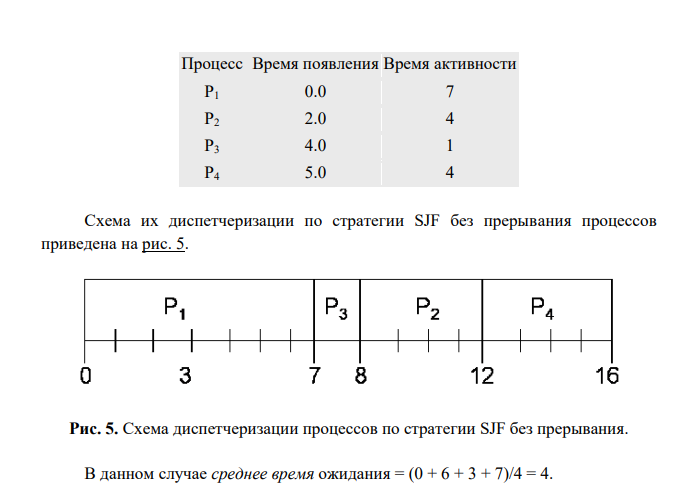
Определить оптимальную стратегию диспетчеризации для среднего времени ожидания и среднего времени выполнения процессов. Рассмотреть стратегии и построить для каждой диаграмму Ганта:

1. **Стратегия First-Come-First-Served (обслуживание в порядке поступления).**



1. **Стратегия Shortest Job First (SJF, обслуживание самого короткого задания первым) – отдельно без прерывания процессов и с прерыванием процессов.**

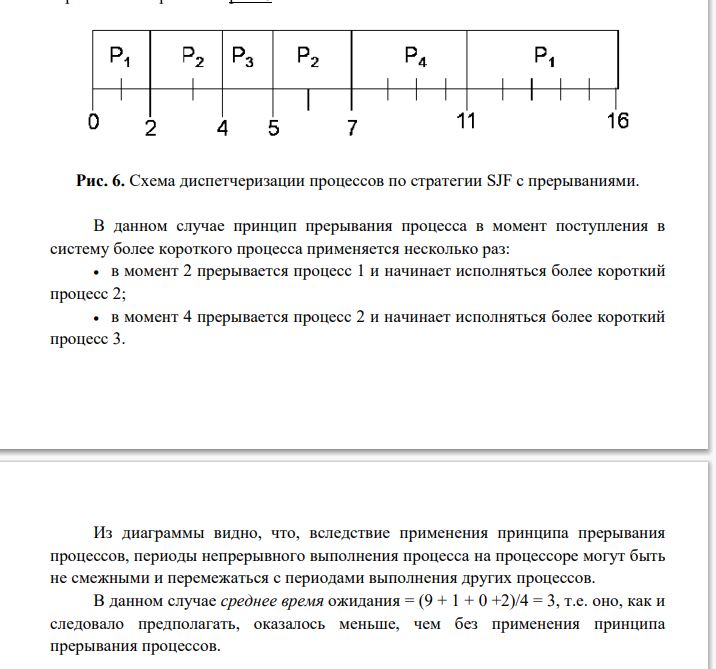
**Без прерывания**



время ожидания =Берётся момент начала процесса и вычитается из него время появления процесса.

время выполнения= время окончания процесса – время появления процесса

**С прерыванием**



время ожидания =Берётся момент начала процесса и вычитается момент появления процесса ,если было прерывание, то необходимо добавить разницу с момента прерывания процесса и момента запуска процесса

время выполнения= время окончания процесса – время появления процесса

1. **По приоритетам - отдельно без прерывания процессов и с прерыванием процессов.**

время ожидания =Берётся момент начала процесса и вычитается момент появления процесса ,если было прерывание, то необходимо добавить разницу с момента прерывания процесса и момента запуска процесса

время выполнения= время окончания процесса – время появления процесса

1. **Стратегия Round Robin (RR, круговая система) для q=4.**

Диаграммы для разных стратегий выполнить на разных листах MS Excel, вывод сформулировать на отдельном листе.

время ожидания= время окончания процесса минус время поступления процесса и минус время выполнения процесса

время выполнения= время окончания процесса – время появления процесса

# **26задание**

1. Дано число размерностью 2 байта, являющееся упакованной датой (в формате год-месяц-день). Распаковать дату, разложив год, месяц и день в разные переменные.

Примечание. год - 7 бит, месяц - 4 бита, день - 5 бит.

1105110=0010101 1001 010112=2B2B16

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Год = 0010101=21+1980=2001 N and 2B67

Месяц = 1001=9 (N and 1E0)>>5

День = 01011=11 (N and FE00)>>9

Ответ: 11 Сентября 2001 года

# **27задание**

1. Дано число размерностью 2 байта, являющееся упакованным времени (в формате часы-минуты-секунды). Распаковать время, разложив год, месяц и день в разные переменные.

Примечание. часы - 5 бит, минуты- 6 бит, секунды- 5 бит.

25456=01100 011011 10000=637016

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Часы = 01100 =12 N and 001F

Минуты= 011011=27 (N and 07E0)>>5

Секунды= 10000=16\*2=32 (N and F800)>>11(B)

Ответ = 12:27:32

# **28 Задание**

Написать программу, которая находит в строке соседей заданного символа и меняет их местами.

mov @str, r0

mov 0, r1

mov 0, r2

mov 0, r3

l:

movb (r0)+, r2 ;взятие символа по адресу

cmp 62, r2 ;проверка на b

jz swap ;если = b

jmp l ;проверка следующего символа

swap:

movb (r0), r3 ; символ после b

sub 2, r0

movb (r0), r1 ; символ перед b

movb r3, (r0) ;запись символа перед b

add 2, r0

movb r1, (r0) ;запись символа после b

stop

str:

data "123b5"