Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рыбинский государственный авиационный технический университет

имени П.А. Соловьева»

Факультет радиоэлектроники и информатики

Кафедра Математическое и программное обеспечение ЭВС

**ОТЧЕТ**

**по лабораторной работе №5**

**R**EADER 1

Студент группы ИПБ-15 Козин О.И .

*(Код) (Подпись, дата) (Фамилия И. О.)*

Рыбинск 2019

Оглавление

[Цель 2](#_Toc3816298)

[Архитектура 3](#_Toc3816299)

[Анализ работы программ 4](#_Toc3816300)

[Приложение. Коды программ. 5](#_Toc3816301)

[Приложение. Интерпретация временных диаграмм 9](#_Toc3816302)

# Цель

1. Разработать архитектуру для задачи Парикмахер
2. Реализовать программу на основе разработанной архитектуры
3. Сделать анализ работы программы

# Архитектура

Client\_in

Barber\_start\_working

Client

Client

Client

Client\_take\_place

Barber\_end\_working

queue

wake

straight

barber

bShop

1

2

3

4

5

6

# Анализ работы программ

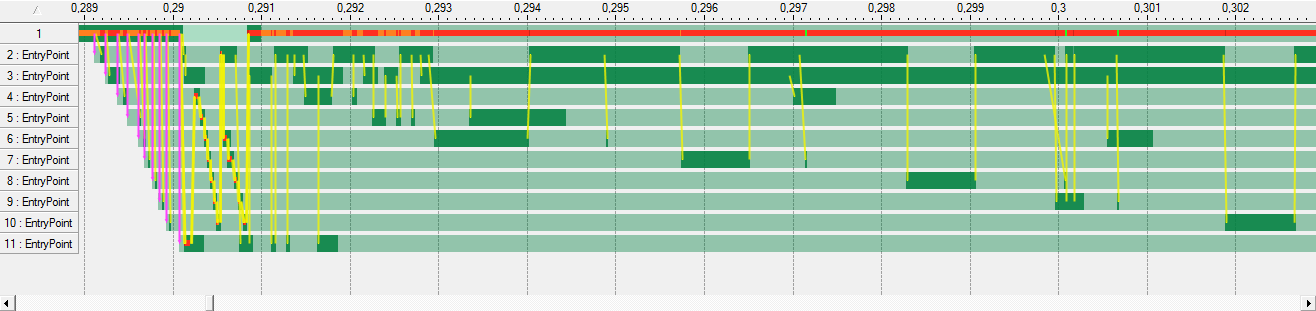


Рис. 1. Временная диаграмма работы программы.

# Приложение. Коды программ.

procedure P01 is

--use Ada.Text\_IO;

use Text\_Io;

subtype my\_type is integer range 100..10000;

package rnd\_choise is

new Random\_Generic (Result\_Subtype => my\_type);

----------------------

task BShop is

--entry client\_work;

--entry barber\_work;

entry client\_in(data: out integer);

entry barber\_Start\_work;

entry barber\_end\_work;

entry client\_take\_place;

end BShop;

task Barber is

entry wake;

entry queue;

entry straight;

--entry B (data : out integer);

end Barber;

task body BShop is

L: integer;

b\_state:integer;

c\_state:integer;

begin

b\_state:=0;

c\_state:=0;

loop

select

accept client\_in (data : out integer) do

Put\_Line("\_\_\_client in");

data:=b\_state;

--if (c\_state = 0)

-- then data:=b\_state;

-- else data:=1;

--end if;

c\_state:=c\_state+1;

end client\_in;

accept client\_take\_place do

put\_line("\_\_\_client\_take\_place");

end client\_take\_place;

or delay 0.00001;

end select;

--delay 3.0;

select

accept barber\_Start\_work do

b\_state:=1;

Put\_Line("Barber start working");

end barber\_start\_work;

or

accept barber\_end\_work do

b\_state:=0;

Put\_Line("Barber stop working & going to sleep----------------------------------------");

end barber\_end\_work;

or delay 0.000001;

end select;

end loop;

end BShop;

------------------------------------------------------------------------------

task body Barber is

client\_queue :integer;

n:my\_type;

l:integer;

begin

loop

client\_queue :=-1;

Put\_Line("sleep");

accept wake do

Put\_Line("wake");

end wake;

Bshop.barber\_start\_work;

loop

select

accept straight do

Put\_Line("working on straight client!");

n:= rnd\_choise.random\_value;

for I in integer range 1..1000 loop

l := i\*i-1;

end loop;

Put\_Line("done with straight client!");

end straight;

or

accept queue do

Put\_Line("working on next client in queue!!");

n:= rnd\_choise.random\_value;

--delay 0.001;

for I in 1..n loop

l := i\*i-1;

--Put\_Line("\*\*\*" & integer'image(l));

end loop;

for I in 1..n loop

l := i\*i-1;

--Put\_Line("\*\*\*" & integer'image(l));

end loop;

for I in 1..n loop

l := i\*i-1;

--Put\_Line("\*\*\*" & integer'image(l));

end loop;

Put\_Line("done with client in queue!!");

Put\_Line("()" & integer'image(queue'count));

client\_queue:=queue'count;

end queue;

or delay 0.0001;

end select;

Put\_Line("/////" & integer'image(client\_queue));

exit when client\_queue = 0;

if (client\_queue = -1)

then client\_queue:=0;

end if;

end loop;

Bshop.barber\_End\_work;

end loop;

end Barber;

------------------

task type Client is

--entry B (data : out integer);

end Client ;

task body Client is

L: integer;

n: my\_type;

begin

loop

Bshop.client\_in(l);

if (l = 0)

then Put\_Line("new client decided to go straight to barber");

barber.wake;

BShop.client\_take\_place;

barber.straight;

else Put\_Line("new client decided to go in queue");

Bshop.client\_take\_place;

barber.queue;

end if;

Put\_Line("client away");

n:= rnd\_choise.random\_value;

for I in 1..100 loop

l := i\*i-1;

end loop;

end loop;

end Client ;

T\_1 : client;

T\_2 : client;

T\_3 : client;

T\_4 : client;

T\_5 : client;

T\_6 : client;

T\_7 : client;

T\_8 : client;

----------------------

begin

Put\_Line("--------");

end P01;

# Приложение. Интерпретация временных диаграмм

