

```

*****
*****
*
*                               Модель СМО (С разделением потока)
*
*****
*****
*
*                               И с х о д н ы е   д а н н ы е
*
*****
*****
E_buf      EQU      9; емкость накопителя (буфера) для П1
t_a        EQU      276; средний интервал между поступающими заявками
t_a_min    EQU      10; минимальный интервал между заявками (не исп. при exp)
t_a_max    EQU      30; максимальный интервал между заявками (не исп. при exp)
t_b        EQU      262; средняя длительность обслуживания заявки в приборе
RN_a       EQU      20; номер генератора для потока
RN_b       EQU      553; номер генератора для длительности обслуживания
* Параметры гиперэкспоненциального распределения:
RN_H       EQU      91; номер генератора для гиперэкспоненциального распределения
qq         EQU      0.3; вероятность выбора первой фазы
tt_1       EQU      600; мат. ожидание первой фазы гиперэкспоненциального
распределения
tt_2       EQU      117; мат. ожидание второй фазы гиперэкспоненциального
распределения
* Параметры гипоекспоненциального распределения (Эрланга):
k_erl     EQU      2; порядок распределения Эрланга
RN_erl1    EQU      31; номер первого генератора для распределения Эрланга 2-го
порядка
RN_erl2    EQU      125; номер второго генератора для распределения Эрланга 2-го
порядка
*****
*****
TU_uzel    TABLE    M1,0.2,0.2,50;
TU_buf     QTABLE     buf1,0.1,0.1,50;
* uzел     STORAGE    2 ; БОЛЬШЕ НЕ ИСПОЛЬЗУЕТСЯ (заменено на SEIZE отдельных
приборов)
Erl_2      VARIABLE   (Exponential(RN_erl1,0,t_a/2))+(Exponential(RN_erl2,0,t_a/2));
сл.величина по закону Эрланга 2-го порядка
*****
* Служебные переменные, необходимые для процедуры GetRandomNumberFromFile *
*****
ErrorCodes MATRIX ,2,1 ; Коды ошибок открытия/закрытия файла
FilePosition MATRIX ,1,1 ; Текущий номер строки в файле
*****
* В качестве исполняемого оставить только ОДИН оператор GENERATE *
*****
GENERATE (Exponential(RN_a,0,t_a))
*GENERATE (GetRandomNumberFromFile("lab1.txt"))
*GENERATE (hyper1(RN_H, qq, tt_1, tt_2))
*GENERATE V$Erl_2

*****
* РАЗДЕЛЕНИЕ ПОТОКА (40% на Прибор 2, 60% идут дальше на Прибор 1)
*****
TRANSFER .400,,Branch2

*****
* ВЕТКА ПРИБОРА 1 (Вероятность 0.6, Очередь = 1)
*****
TEST      L           Q$buf1,E_buf,zux ; Проверка очереди (если
занято - на выход zux)

```

```

        QUEUE      buf1
        SEIZE       Pribor1          ; Занимаем 1-й прибор
        DEPART      buf1
        ADVANCE      (Exponential(RN_b,0,t_b))
        RELEASE     Pribor1
        TRANSFER     ,CommonExit      ; Уходим на сбор статистики

*****
* ВЕТКА ПРИБОРА 2 (Вероятность 0.4, Очередь = 0)
*****
Branch2  GATE      NU              Pribor2,зух ; Проверка занятости (если занят -
сразу на выход зух)
        SEIZE       Pribor2          ; Занимаем 2-й прибор
        ADVANCE      (Exponential(RN_b,0,t_b))
        RELEASE     Pribor2

*****
* ВЫХОД И СБОР СТАТИСТИКИ
*****
CommonExit TABULATE              TU_uzel
        TERMINATE   1

*****
* ПОТЕРЯННЫЕ ЗАЯВКИ
*****
зух      TERMINATE   1

*****
*****
* Процедура GetRandomNumberFromFile (без изменений)
*****
*****
PROCEDURE GetRandomNumberFromFile(FileName) BEGIN
    TEMPORARY OpenError, CloseError, LineFromFile, FileId;
    FileId = 1;
    OpenError = open(FileId,FileName);
    if (OpenError /= 0) then begin
        FileId = 2;
        OpenError = open(FileId,FileName);
        if (OpenError /=0) then begin
            ErrorCodes[1,1] = OpenError;
            return "";
        end;
    end;
    FilePosition[1,1] = FilePosition[1,1] + 1;
    seek(FileId,FilePosition[1,1]);
    LineFromFile = read(FileId);
    if (LineFromFile = "") then begin
        FilePosition[1,1] = 1;
        seek(FileId,FilePosition[1,1]);
        LineFromFile = read(FileId);
    end;
    CloseError = close(FileId);
    if (CloseError /=0) then begin
        ErrorCodes[2,1] = CloseError;
        return "";
    end;
    return value(LineFromFile);
END;
*****
* Процедура hyper1 (без изменений)

```