```
else{
    erro_timeout = true;
    return (0);
}
```

(R. o código ficará esperando, em intervalos de 10ms, até que chegue algum dado pela serial do PIC ou que a variável tempo não seja maior do que 500. Caso chegue algum dado pela serial, ou o tempo seja maior do que 500, o escopo sai do while. É testado se chegou algo na serial. Se positivo, é chamada a função getc() para fazer a leitura do dado na serial. Caso contrário, erro_timeout recebe true como tratamento de estouro de timeout.)

13. Explique o escopo de código abaixo.

```
WHILE TRUE
  SEQ
    ALT
      call ? new_voltage
        -- outras ações
      clock ? time
        -- ação para timeout
```

(R. O comando seq executará sequencialmente a estrutura de código que a segue. O comando alt executará o escopo que o segue paralelamente. Assim, será executada a função de recepção de tensão em paralelo com a função de contagem de tempo (timeout). Caso a tensão seja lida antes do estouro de tempo, o escopo logo abaixo de call será executado. Caso contrário, a escopo logo abaixo da função clock será executada.)

14. Diferencie processo periódico de processo esporádico. (R. os processos periódicos amostram dados ou executam um loop de controle e têm um prazo de encerramento explícito que deve ser atendido. Já os processos esporádicos não amostram dados que são gerados periodicamente e não têm prazos de encerramento específicos.)

15. Cite dois atributos de um escopo temporal. (R. Deadline: tempo no qual a execução do TS deve ser finalizada; Mínimo delay: o mínimo tempo que deve transcorrer antes do início da execução do TS.)

16. Explique a característica de um sistema operacional pré-emptivo. (R. Em um SO pré-emptivo, um processo em execução pode ser interrompido para execução de outro processo de prioridade maior.)

LISTA DE EXERCÍCIO DE SOFTWARE DE TEMPO-REAL

1. **Conceitue ação atômica.**

R. Uma ação é atômica se o processo desempenhando a ação não se preocupa com a existência de qualquer outro processo ativo, e nenhum outro processo ativo se preocupa com a atividade do processo durante o tempo que o processo está desempenhando a ação.

2. **Implemente uma ação atômica utilizando o conceito de exclusão mútua.** (

R . wait(mutual_exclusion_semaphore)

atomic_action

signal(mutual_exclusion_semaphore)

3. **No escopo de código abaixo, explique a finalidade das funções Allocate e Free.**

```
Package RESOURCE_MANAGER is

  type RESOURCE is private;

  function ALLOCATE return RESOURCE;

  function FREE (THIS_RESOURCE:RESOURCE)

Private

  type RESOURCE is ...

End RESOURCE_MANAGER;
```

(R. A função allocate é utilizada para reservar o recurso encapsulado pelo gerenciador de recursos. Outro processo executando um allocate sobre o recurso já alocado ficará em estado suspenso até que o recurso seja liberado. A função free libera o recurso alocado. Isso fará com que um processo suspenso sobre o recurso, saia deste estado e possa utilizar o recurso.)

4. **Conceitue deadlock.** (R. Um estado do sistema em que um ou mais processos ficam em estado suspenso por tempo indeterminado.)

5. **No exemplo abaixo, verifique se há deadlock, e explique sua resposta.**

P1	*P2*
Main()	*Main()*
allocate (R1)	*allocate (R2)*
allocate (R2)	*allocate (R1)*

(R. Há deadlock sim. Utilizando a análise por interleaving, o processo P1 aloca o recurso R1. Posteriormente, o processo P2 aloca o recurso R2. Depois, o processo P1 tenta alocar R2, que já está alocado por P2. Então, entra em estado suspenso. O mesmo acontece com o processo P2, que tenta alocar o recurso R1, já alocado. Assim, os dois processos ficam em estado suspenso.)

06. Cite duas maneiras de se ter acesso ao clock em um sistema de tempo-real. (R. Implementando funções de acesso ao clock na linguagem; Implementando um driver de dispositivo de clock associado aos processadores ou microcontroladores.

07. Qual a finalidade do escopo de código em OCCAM abaixo?

```
TIMER clock:
INT old, new, interval:
SEQ
  clock ? Old
  other functions
  clock ? new
  interval := new MINUS old
```

(R. A variável old recebe a leitura do relógio de tempo-real, Posteriormente, são executadas outras funções. A variável new recebe a leitura do relógio de tempo-real. E a variável interval recebe a diferença entre as variáveis old e new. Assim, interval representa o intervalo de tempo de execução das funções entre old e new.)

08. Qual a desvantagem do atraso de um processo que utiliza o escopo abaixo?

```
NOW:= CLOCK;
Loop
  exit when (clock-NOW) > 10.0;
End Loop;
```

(R. A desvantagem é que o processo atrasa em execução, aumentando o processamento do sistema.)

09. Qual a vantagem de um processo que executa a função abaixo utilizando o Suporte de Run-Time da linguagem?

```
delay 10.0; espera dez segundos
```

(R. A vantagem é que o processo atrasa em estado suspenso, não comprometendo o processamento do sistema.)

10. Conceitue timeout. (R. Um timeout é uma restrição no tempo de um processo que está esperando por um evento.)

11. Cite uma finalidade no uso de timeout. (R. detecção de falhas na passagem de mensagens entre processos.)

12. Explique o escopo de código abaixo.

```
while(!kbhit() && (++tempo<500) delay_ms(10);
if (kbhit()) return (getc());
```