

Questionário - 20

Arthur C. M. Barcella e Matheus P. Salazar

Explique o que é e como funciona o tratamento de interrupções em dois níveis pelo sistema operacional.

O tratamento de interrupções em dois níveis é utilizado no sistema operacional quando precisa resolver casos com tempo não mapeado. Os dois níveis são:

1 - Nível de hardware: A interrupção ocorre a partir de um sinal elétrico a ser enviado para a CPU, indicando uma mudança física no hardware, por exemplo o aperto de uma das teclas do teclado.

2 - Nível de software: No nível de software ocorre o tratamento da interrupção recebida, ou seja, quando a interrupção do teclado chegou a CPU por exemplo, a CPU escreve o caractere na tela, então essa é a ação gerada em software.

Considere um dispositivo físico cujo controlador tem quatro portas de 8 bits:

porta	sentido	função	valor
<i>data_in</i>	<i>dev</i> \rightarrow <i>cpu</i>	leitura de dados	byte lido
<i>data_out</i>	<i>cpu</i> \rightarrow <i>dev</i>	escrita de dados	byte a escrever
<i>status</i>	<i>dev</i> \rightarrow <i>cpu</i>	estado do dispositivo	IDLE, BUSY ou ERROR
<i>control</i>	<i>cpu</i> \rightarrow <i>dev</i>	operação a efetuar	READ, WRITE ou RESET

Essas portas são acessadas através das operações “*val=in(port)*” e “*out(port,val)*”. Escreva em pseudocódigo as funções “*char readchar()*” e “*void writechar(char c)*” do driver desse dispositivo, que implementam respectivamente as operações de leitura e de escrita de caracteres individuais, usando a estratégia de polling (interação por programa).

```
1 // Autor: Arthur Cadore M. B.
2 // SOP - 2023.1
3
4 // Declaração da porta de entrada e porta de saída
5 port inPort;
6 port outPort;
7
8 // Função para ler um caractere da porta de entrada
9 char readchar() {
10     // Variável para armazenar o caractere lido
11     char character;
12
13     // Loop de polling até que haja um caractere disponível na
14     // porta de entrada
15     while (in(inPort) == 0) {
16         // Aguarda até que um caractere esteja disponível
17     }
18
19     // Lê o caractere da porta de entrada
20     character = in(inPort);
21
22     // Retorna o caractere lido
23     return character;
24 }
25
```

```
26 // Função para escrever um caractere na porta de saída
27 void writechar(char c) {
28     // Loop de polling até que a porta de saída esteja pronta
29     // para receber um caractere
30     while (in(outPort) != 0) {
31         // Aguarda até que a porta de saída esteja livre
32     }
33
34     // Escreve o caractere na porta de saída
35     out(outPort, c);
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

Considerando o dispositivo da questão anterior, escreva em pseudocódigo a função “char readchar()” do driver desse dispositivo, que implementa a operação de leitura de caracteres individuais, usando a estratégia de interação por eventos (interrupções).

```
1 // Autor: Arthur Cadore
2 // SOP - 2023.1
3
4 // Declaração da porta de entrada e porta de saída
5 port inPort;
6 port outPort;
7
8 // Variável global para armazenar o caractere lido
9 char character;
10
11 // Função de manipulação de interrupção para ler um caractere da porta de entrada
12 void interrupter() {
13     // Lê o caractere da porta de entrada
14     character = in(inPort);
15 }
16
17 // Função para configurar a interrupção de leitura na porta de entrada
18 void interrupt_Init() {
19     // Configura a interrupção para ocorrer quando um caractere estiver disponível na porta de entrada
20     // A função interrupter() será chamada quando a interrupção for acionada
21     // Aqui devemos desenvolver o modo como a interrupção irá ocorrer para o sistema operacional, ou seja, pode variar de sistema para sistema.
22 }
23
24 // Função para ler um caractere da porta de entrada
25 char readchar() {
26     // Aguarda a ocorrência de uma interrupção que indica que um caractere está disponível
27     // A função interrupt_Init() deve ser chamada antes de chamar a função readchar() para configurar corretamente a interrupção
28     // com a interrupção já configurada, chamamos a função interrupter para coletar o caractere.
29     interrupter();
30
31     // Retorna o caractere lido
32     return character;
33 }
```

Questionário - 20

Arthur C. M. Barcella e Matheus P. Salazar