

Kernel mode

Lista 01 - Sistemas Operacionais - Sophia Carrazza

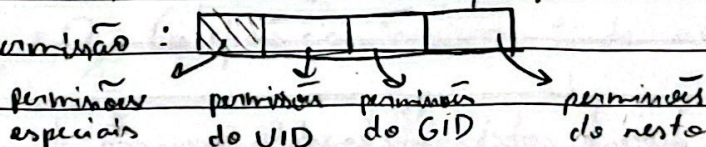
1- As instruções que só deveriam ser permitidas em modo supervisor são:

- a - Desabilitar todas as interrupções, pois isso influencia no controle do hardware e no funcionamento do SO.
- c - Alterar o relógio da máquina, já que o timer da máquina é uma instrução privilegiada, alterar esse relógio é no supervisor.
- d - Mudar o mapa de memória, se forem áreas não autorizadas ao user padrão, essenciais à segurança do SO, deve ser executado apenas em modo supervisor.

2- uid = user ID / gid = group ID → $\begin{matrix} \text{gid} \\ \text{uid} = 12 \\ \text{uid} = 6 \end{matrix}$ → $\begin{matrix} \text{rwx} & \text{r-x} & \text{---} \\ \text{UID} & \text{GID} & \text{resto dos usuários} \end{matrix}$

• leitura = r / escrita = w / execução = x
(alterar) (executar comandos e arquivos) (ler, pesquisar e executar)

• Modo de permissão:



* Resposta = O usuário com uid = 6 poderá alterar o arquivo, já que ele possui o mesmo GID do dono do arquivo, e a permissão do grupo é de leitura e execução (r-x). Porém, ele não poderá realizar ações de escrever, como o usuário de UID = 12 pode.

3- SIGALRM é um sinal, em Linux, de número 14, que é enviado a um processo quando um timer é setado depois da função alarm().
expirar (uma função de temporizador em C)

Assim, para enviar um SIGALRM a um processo, utilizamos a função alarm() em C, com o parâmetro de segundos até gerar o sinal (ex: alarm(10) gera o sinal em 10 segundos).

É possível ignorar o SIGALRM, mas isso pode alterar o desempenho de alguns processos do SO e causar comportamentos prejudiciais, já que funções podem continuar rodando ininterruptamente.

1/1

$$10 \cdot 10 \cdot \frac{15}{10}$$
$$10 \cdot 10 \cdot 1,5$$

$$10 \cdot 15 =$$

S T Q Q S S

4- Um computador com chaveamento de processo feita por hardware necessita de informações constantemente armazenadas dos registradores, da pilha de processos (ou do registro que armazena o estado dos processos) e do controlador que gerencia o chaveamento.

Assim, esse chaveamento poderia ser feito por meio da detecção dos processos, o armazenamento do estado atual na CPU, execução de um novo processo e reinicialização do estado, e a continuação disso em loop.

5- Cada tarefa representa uma thread, que é executada simultaneamente junto a outras. Assim, nesse sistema, existe uma pilha para cada thread, já que cada thread precisa de uma pilha para armazenar estados e variáveis locais, sem interferência uma na outra.

Se existisse uma pilha por processo, poderiam ocorrer conflitos de dados e de espaços.

6- T = segundos que o processo precisa p/ completar operação

n = min de processos concorrentes.

▷ Ele trabalha c/ até 10 processos, usando escalonamento.

▷ Ele executa 1 instrução de cada processo por vez, em sequência.

Logo, se n for menor ou igual a 10, o tempo para um processo será $T \cdot n$. Se n for maior que 10, será $T \cdot 10 \cdot \left\lceil \frac{n}{10} \right\rceil$.

(escalonamento c/ 1
instrução por vez)

somente o
pode ser executado
por vez

7- Se um processo aparecer mais de uma vez em uma lista de processos, ele será priorizado na sequência, sendo selecionado com mais frequência e alocando mais carga/memória a ele. Assim, isso é útil se um processo é mais importante ou exige mais tempo de execução (assim ele pode finalizar no mesmo tempo dos demais ao invés de permanecer sozinho após o término de todos os demais).

spiral

Quantum → Intervalo máximo de tempo que um processo pode executar continuamente antes que o escalador interrompa o processo e passe p/ o próximo processo.

Quantum → Intervalo máximo de tempo que um processo pode executar continuamente antes que o escalador interrompa o processo e passe p/ o próximo processo.

8- eficiência = $\frac{\text{Tempo útil CPU}}{\text{Tempo total CPU}}$ (executando algum processo)
 $\Rightarrow \frac{p/\text{round}}{\text{vobim}} = \frac{Q}{Q+S} \rightarrow \text{overhead}$

a) eficiência = $\frac{\alpha}{\alpha+S}$ ($\alpha = T$)

b) $Q > T \rightarrow$ (o tempo nunca será interrompido, ele roda por inteiro)
 \hookrightarrow eficiência = $\frac{T}{T+S}$

c) $S < Q < T$ (o tempo é maior que o quantum e o overhead)
 eficiência = $\frac{Q}{Q+S+(T-Q)}$

d) $Q < T$ $\frac{Q}{Q+S} \Rightarrow \frac{S}{S+S} \Rightarrow \frac{1}{2}$
 \hookrightarrow o quantum não é suficiente, mas o overhead é.

e) $Q \rightarrow 0$
 $\hookrightarrow = \frac{Q}{Q+S} = 0$ (a eficiência também tende a 0)

9-

| Processo | Tempo de Exec. | Prioridade |
|----------------|----------------|------------|
| T ₁ | 10 | 3 |
| T ₂ | 6 | 5 |
| T ₃ | 2 | 2 |
| T ₄ | 4 | 1 |
| T ₅ | 8 | 4 |

(sem prioridade)

a) Round-Robin: (quantum = 5)

$\rightarrow T_1 - 5 / \text{falta } 5^{\circ} (T=5)$ $T_1 - +5 / \text{concluiu } (T=26)$
 $T_2 - 5 / \text{falta } 1^{\circ} (T=10)$ $T_2 - +1 / \text{concluiu } (T=27)$
 $T_3 - 2 / \text{concluiu } 2^{\circ} (T=12)$ $T_5 - +3 / \text{concluiu } (T=30)$
 $T_4 - 4 / \text{concluiu } 1^{\circ} (T=16)$
 $T_5 - 5 / \text{falta } 3^{\circ} (T=21)$
 ciclo 2
 ciclo 1
 Turnaround = 22,2 min/ps
 média

spiral

_ / _ / _

$$1 \quad 2 \quad 3 \quad 4 \quad 5$$

$$T_4 \rightarrow T_3 \rightarrow T_1 \rightarrow T_5 \rightarrow T_2$$

b) Escalonamento com Prioridade:

- $T_4 \rightarrow 4 / \text{concluir} (T=4)$
- $T_3 \rightarrow 2 / \text{concluir} (T=6)$
- $T_1 \rightarrow 10 / \text{concluir} (T=16)$
- $T_5 \rightarrow 8 / \text{concluir} (T=24)$
- $T_2 \rightarrow 6 / \text{concluir} (T=30)$

ou seja:

$$\begin{array}{r} 30 \\ + 24 \\ \hline 16 \\ + 6 \\ \hline 4 \\ \hline 80 / 5 = 16 \end{array}$$

Turnaround = 16 minutos
médio

c) FCFS (first come, first-served) sem interrupções

- $T_1 \rightarrow 10 / \text{concluir} (T=10)$
- $T_2 \rightarrow 6 / \text{concluir} (T=16)$
- $T_3 \rightarrow 2 / \text{concluir} (T=18)$
- $T_4 \rightarrow 4 / \text{concluir} (T=22)$
- $T_5 \rightarrow 8 / \text{concluir} (T=30)$

ou seja:

$$\begin{array}{r} 30 \\ + 22 \\ \hline 18 \\ + 16 \\ \hline 10 \\ \hline 96 / 5 = 19,2 \end{array}$$

Turnaround = 19,2 minutos
médio

(shortest job first)

d) SJF (quantum = 5) e uma tarefa executada por vez

- $T_3 \rightarrow 2 / \text{concluir} (T=2)$
- $T_2 \rightarrow 1 / \text{concluir} (T=16)$
- $T_4 \rightarrow 4 / \text{concluir} (T=6)$
- $T_5 \rightarrow 3 / \text{concluir} (T=19)$
- $T_2 \rightarrow 5 \text{ resta } 1 (T=7)$
- $T_1 \rightarrow 5 / \text{concluir} (T=24)$
- $T_5 \rightarrow 5 \text{ resta } 3 (T=10)$
- $T_1 \rightarrow 5 \text{ resta } 5 (T=15)$

ciclo 1

ciclo 2

$$\begin{array}{r} 24 \\ + 19 \\ \hline 16 \\ + 6 \\ \hline 2 \\ \hline 67 / 5 = 13,4 \end{array}$$

Turnaround = 13,4 minutos
médio

10- Evento Período (ms) Demanda (ms)

| | | |
|---|-----|----|
| 1 | 50 | 35 |
| 2 | 100 | 20 |
| 3 | 200 | 10 |
| 4 | 250 | X |

$$U \rightarrow \frac{35}{50} + \frac{20}{100} + \frac{10}{200} + \frac{X}{250} < 1$$

$$0,7 + 0,2 + 0,05 + \frac{X}{250} < 1$$

→ p/ continuar escalonável, a utilização da CPU não pode passar de 100%.

$$(-) 1 + 0,7 - 0,2 - 0,05 > \frac{X}{250}$$

$$+ 0,05 > \frac{X}{250}$$

Resp = O maior valor possível

$$X = 12,5$$

p/ X deve ser 12,5 ms.

spiral