



Iniciado em domingo, 20 jun 2021, 18:07

Estado	Finalizada
--------	------------

Concluída em domingo, 20 jun 2021, 18:38

Tempo 31 minutos 9 segundos
empregado

Questão 1

Incorreto

Atingiu 0,00
de 1,00

Assinale a alternativa CORRETA.

Escolha uma opção:

- ☒ a. Um processo pode criar outro processo, chamado processo filho, o qual possui um PID diferente, mas o mesmo código e os mesmos endereços do processo pai.
- ☐ b. O espaço de endereçamento do processo filho é uma cópia do espaço do processo pai, sendo que eles normalmente compartilham o segmento relativo ao código do programa.
- ☐ c. Se um processo pai morre, todos os seus processos filhos morrem também.
- ☐ d. Por se tratar de uma cópia, o processo filho tem as mesmas threads que o processo pai.
- ☐ e. Assim como um processo pai pode ter vários filhos, um processo filho pode ser configurado para ter vários pais. Isto, porém, é um caso especial, usado apenas para prevenir a ocorrência de processos zumbis.

A resposta correta é: O espaço de endereçamento do processo filho é uma cópia do espaço do processo pai, sendo que eles normalmente compartilham o segmento relativo ao código do programa.

Histórico de respostas

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:09	Salvou: Um processo pode criar outro processo, chamado processo filho, o qual possui um PID diferente, mas o mesmo código e os mesmos endereços do processo pai.	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Incorreto	0,00



Escolha uma opção:

- ☐ a. A taxa de ocupação da CPU tem relação com o número de operações de E/S realizadas.
- ☐ b. A probabilidade de espera por operações de E/S aumenta conforme aumenta o grau de multiprogramação.
- ☐ c. Modelos de multiprogramação otimistas são irrealistas porque todo processo realiza operações de E/S.
- ☐ d. O número de programas que uma CPU pode executar por vez não depende das operações de E/S.
- ☒ e. Em um sistema multiprogramado não podemos assumir nenhuma temporização fixa para computar o tempo de execução.

A resposta correta é: Modelos de multiprogramação otimistas são irreais porque todo processo realiza operações de E/S.

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:10	Salvou: Em um sistema multiprogramado não podemos assumir nenhuma temporização fixa para computar o tempo de execução.	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Incorreto	0,00

Atingiu 0,00
de 1,00

Escolha uma ou mais:

- ☐ a. Um processo pode ir do estado de "blocked" para "ready".
- ☒ b. Um processo pode ir do estado de "blocked" para "running".
- ☐ c. Um processo pode ir do estado de "running" para "ready".
- ☒ d. Um processo pode ir do estado de "running" para "blocked".
- ☐ e. Um processo pode ir do estado de "ready" para "blocked".
- ☒ f. Um processo pode ir do estado de "ready" para "running".

As respostas corretas são: Um processo pode ir do estado de "blocked" para "ready"., Um processo pode ir do estado de "running" para "ready"., Um processo pode ir do estado de "running" para "blocked"., Um processo pode ir do estado de "ready" para "running"

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:21	Salvou: Um processo pode ir do estado de "blocked" para "running"; Um processo pode ir do estado de "running" para "blocked".; Um processo pode ir do estado de "ready" para "running"	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Incorreto	0,00

Escolha uma opção:

- ☐ a. Um processo é uma abstração provida pelo S.O de forma permitir na multiprogramação
- ☒ b. Devido ao chaveamento entre processos, o tempo de computação requerido para um mesmo programa pode variar.
- ☐ c. Além dos S.O.s com base no kernel do Linux, apenas os sistemas Windows implementam, de fato, o conceito de processo.
- ☐ d. Duas instâncias de um mesmo programa são considerados dois processos, com pelo menos uma thread cada.
- ☐ e. Para simplificar o modelo de processos, cada processo considera que tem acesso exclusivo à CPU, de forma que podemos assumir a existência de uma CPU "virtual" para cada processo.

A resposta correta é: Além dos S.O.s com base no kernel do Linux, apenas os sistemas Windows implementam, de fato, o conceito de processo.

Histórico de respostas

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:22	Salvou: Devido ao chaveamento entre processos, o tempo de computação requerido para um mesmo programa pode variar.	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Incorreto	0,00

Questão 7

Incorreto

Atingiu 0,00 de 1,00

Considere um ambiente de multiprogramação com uma única CPU, com t sendo a taxa de 100% de uso da CPU e p o número de processos em execução. Assim, por que não podemos considerar que o uso de CPU por cada processo ocorre a uma taxa de t/p ?

Escolha uma opção:

- ☒ a. Porque os processos não são iguais, realizando diferentes operações.
- ☐ b. Porque a CPU nunca é usada 100% do tempo.
- ☐ c. Porque o cálculo correto é p/t , uma vez que o número de processos é maior do que o de CPUs
- ☐ d. Porque, além do tempo de cada processo, há um tempo gasto pelo S.O. para, dentre outras coisas, realizar trocas de contexto.
- ☐ e. Nenhuma das respostas anteriores

A resposta correta é: Porque, além do tempo de cada processo, há um tempo gasto pelo S.O. para, dentre outras coisas, realizar trocas de contexto.

Histórico de respostas

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:35	Salvou: Porque os processos não são iguais, realizando diferentes operações.	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Incorreto	0,00

Escolha uma ou mais:

- ☐ a. As threads não visíveis ao S.O.
- ☒ b. A criação de uma thread é mais rápida que a de um processo
- ☐ c. Troca de contexto entre threads é mais rápida que entre processos
- ☐ d. O espaço de endereçamento das threads é menor que o dos processos
- ☐ e. Cada thread tem a sua própria pilha.

As respostas corretas são: A criação de uma thread é mais rápida que a de um processo, Troca de contexto entre threads é mais rápida que entre processos

Histórico de respostas

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:38	Salvou: A criação de uma thread é mais rápida que a de um processo	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Parcialmente correto	0,50

Escolha uma ou mais:

- As respostas corretas são: Threads em modo usuário não são visíveis ao S.O., O run-time system não é necessário para threads em modo kernel, Uma das vantagens das threads em modo usuário é permitir que o escalonamento seja personalizado, Threads no modo usuário devem ceder o controle da CPU voluntariamente

Passo	Hora	Ação	Estado	Pontos
1	20/06/2021 18:07	Iniciada	Ainda não respondida	
2	20/06/2021 18:38	Salvou: Uma das vantagens das threads em modo usuário é permitir que o escalonamento seja personalizado; Threads no modo usuário devem ceder o controle da CPU voluntariamente	Resposta salva	
3	20/06/2021 18:38	Tentativa finalizada	Parcialmente correto	0,50

