

- 1) O que é um processador *multithread*?
- a) Um processador que executa apenas uma thread por vez, priorizando a eficiência de execução.
 - b) Um processador que suporta a execução de múltiplas threads simultaneamente.
 - c) Um processador projetado especificamente para executar operações matemáticas complexas.
 - d) Um processador que utiliza técnicas de paralelismo de instrução para melhorar o desempenho.
- 2) O IFC dispõe de um sistema operacional chamado IFC OS. Esse sistema não oferece formas de processamento concorrente. Cite e explique três complicações importantes que o processamento concorrente acrescentaria no IFC OS, caso ele dispusesse desse recurso.
- 3) Suponha que o arquivo executável chamado `processos` foi gerado a partir do código abaixo. Qual será o resultado exibido na saída padrão (`stdout`) se o arquivo for executado na linha de comando (`./processos`)?

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main() {

    fork();
    fork();

    printf("Texto\n");

    return 0;
}
```

- a) Texto
- b) Texto
 Texto
- c) Texto
 Texto
 Texto
- d) Texto

Texto

Texto

Texto

e) Texto\n

4) (Prefeitura Municipal de Caranaíba – 2019 – Analista de Tecnologia da Informação)
Um processo em um Sistema Operacional pode estar em um dos estados: novo, pronto, executando, esperando e terminado. São transições de estados de um processo, EXCETO de

- a) esperando para pronto.
- b) executando para pronto.
- c) executando para esperando.
- d) esperando para executando.

5) Suponha que um programa seja 80% não paralelo e calcule o seu *speedup* considerando que ele será executado em um sistema com

- a) dois núcleos de processamento.
- b) quatro núcleos de processamento.

6) Considere a existência de quatro processos, P_1 , P_2 , P_3 e P_4 , com surtos de CPU de 6 ms, 8 ms, 7 ms e 3 ms, respectivamente. Considere ainda que todos os processos chegaram no tempo zero, exatamente na ordem em que foram escritos. Assinale a alternativa correta:

- a) Utilizando FCFS, o tempo médio de espera será de 11,25 ms.
- b) O throughput do algoritmo FIFO será maior se comparado ao do algoritmo SJF.
- c) Um escalonamento com filas multinível e quantum de 8 ms poderia fazer com que os quatro processos fossem executados simultaneamente.
- d) O tempo médio de espera será de 7 ms se o algoritmo SJF for utilizado.
- e) Se o algoritmo Round-Robin for utilizado, independentemente do *quantum* escolhido, ocorrerá *starvation*.

7) Assumindo que o escalonador utilize o algoritmo preemptivo de fila de prioridades para definir o processo que ganha o processador, calcule o tempo médio de espera considerando o seguinte conjunto de processos, com prioridade e tempo de CPU (dado em milissegundos). Assuma que todos os processos estão prontos no tempo zero, na ordem indicada na tabela, e que o algoritmo FCFS é usado em caso de empate de prioridades.

Processo	Tempo (em ms)	Prioridade
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

8) Suponha que os processos P_1 , P_2 e P_3 chegaram ao mesmo instante e que o surto de CPU de P_1 é 24 ms, de P_2 é 3 ms e de P_3 é 3 ms. Calcule o throughput e o tempo médio de espera considerando que foi utilizado o algoritmo de escalonamento

- a) FCFS
- b) SJF

9) (adaptado de DPE-RS – 2017 – Analista – Tecnologia da Informação) Dentre as políticas de escalonamento de processos a seguir, a que apresenta maior probabilidade de ocasionar o starvation é a

- a) Round Robin.
- b) de tempo compartilhado.
- c) de filas multinível com mecanismo de aging
- d) preemptiva.
- e) não preemptiva.

10) O que é uma condição de corrida (*race condition*) em programação paralela?

- a) Um estado em que duas ou mais threads tentam modificar um recurso compartilhado ao mesmo tempo, levando a resultados inconsistentes.
- b) Uma situação em que uma thread bloqueia a execução de outra thread indefinidamente.
- c) Uma condição em que o tempo de execução de uma tarefa excede um limite predefinido.
- d) Uma situação em que uma thread é interrompida antes de concluir sua execução.

11) (DPE/AM – 2018 – Assistente Técnico – Defensoria – Assistente Técnico – Suporte)

Em um sistema operacional típico de um computador, três processos se encontram na seguinte situação:

- o processo P_1 envia uma mensagem ao Processo P_2 .

- o processo P2, ao receber a mensagem de P1, responde essa mensagem a P1.
- o processo P1, ao receber a mensagem de resposta de P2, responde a P2 com uma nova mensagem, e assim sucessivamente.
- o processo P3 se encontra bloqueado, situação da qual sairá apenas quando receber uma mensagem do processo P1.

Considerando que a prioridade do processo P3 é menor do que as prioridades dos processos P1 e P2, tem-se que

- a) a cada troca de mensagens entre P1 e P2 as respectivas prioridades automaticamente serão reduzidas, e quando elas forem inferior à do processo P3, esse será executado.
- b) após passar 1 segundo bloqueado, o relógio de tempo real do sistema operacional automaticamente dará oportunidade de o processo P3 ser desbloqueado.
- c) o processo P3 sairá dessa situação assim que uma interrupção qualquer ocorra.
- d) ocorrerá uma situação conhecida como impasse (deadlock).
- e) ocorrerá uma situação conhecida como inanição (starvation).

12) (Quadrix – CRQ 18º Região/PI – 2016 – Auxiliar Técnico de Informática) Quantos processos, no mínimo, podem estar envolvidos em um deadlock de um sistema operacional?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5

13) Qual é o método de comunicação mais adequado para a comunicação entre processos em diferentes máquinas?

- a) Pipes.
- b) Sockets.
- c) Memória compartilhada.
- d) Semáforos.

14) O que é um sistema de memória compartilhada em programação paralela?

- a) Um sistema que permite que várias threads acessem a mesma área de memória simultaneamente.
- b) Um sistema que divide a memória principal entre várias threads para melhorar o desempenho.

- c) Um sistema que utiliza memória cache compartilhada entre os processadores.
- d) Um sistema que sincroniza o acesso à memória entre várias threads para evitar conflitos.

15) (POSCOMP 2019 – Questão 45) Considere o programa abaixo escrito em linguagem C. No instante da execução da linha 5, ter-se-á uma hierarquia composta de quantos processos e threads, respectivamente?

1	main() {
2	int i;
3	for (i=0; i<3; i++)
4	fork();
5	while(1);
6	}

- a) 1 e 0.
- b) 3 e 0.
- c) 4 e 1.
- d) 7 e 7.
- e) 8 e 8.

16) (adaptado de UFRN – 2018 – Engenheiro da Computação) Sistemas operacionais modernos têm uma gerência de processos e de threads bem definida. Nesse contexto, é correto afirmar que

- a) threads de um mesmo processo compartilham o mesmo contador de programa (PC – *program counter*).
- b) threads de um mesmo processo compartilham a mesma seção de código na memória.
- c) threads de um mesmo processo compartilham a mesma seção da pilha na memória.
- d) todas as variáveis de uma thread são compartilhadas com as outras threads do mesmo processo.
- e) todos os contextos de uma thread são compartilhados com as outras threads do mesmo processo.

17) (CESPE/CEBRASPE – EMBASA – 2010 – Analista de Saneamento – Analista de Tecnologia da Informação - Redes) Entre as atividades típicas de gerência de processos de um sistema operacional, estão fornecer mecanismos para a

sincronização e comunicação de processos, e suspender e retomar processos.
Certo ou errado?

18) (VUNESP – EsFCEEx - 2020 – Oficial – Informática) Os semáforos são utilizados em sistemas operacionais e permitem controlar o acesso a recursos compartilhados. Uma das suas características essenciais é a atomicidade, que garante que

- a) haja a preservação do valor do semáforo mesmo em casos de reinicialização do sistema.
- b) o valor máximo que um semáforo possa receber nunca seja ultrapassado.
- c) uma vez terminado o uso de um semáforo, ele seja restaurado em suas características originais para uso futuro.
- d) uma vez iniciada uma operação de um semáforo, nenhum outro processo pode ter acesso a ele até que essa operação tenha acabado ou tenha sido bloqueada.
- e) as operações relacionadas aos semáforos possam ser executadas com segurança, mesmo que tenham que ser interrompidas antes dos respectivos términos.

19) (POSCOMP 2006, Questão 32) Qual dos seguintes mecanismos é o menos recomendado para se implementar regiões críticas em sistemas operacionais?

- a) Semáforo
- b) Espera ocupada
- c) Troca de mensagens
- d) Monitores
- e) Variáveis de condição

20) (POSCOMP 2015, Questão 45) Em um sistema operacional multitarefa, três processos compartilham dois recursos. Cada um destes processos possui, no mínimo,

- a) seis seções críticas.
- b) quatro seções críticas.
- c) três seções críticas.
- d) duas seções críticas.
- e) uma seção crítica