

Primeira Lista de Exercícios de SO (5 pontos)

1) O que é um Sistema Operacional?

É o software que controla os dispositivos de hardware.

2) Quais são os principais objetivos de um Sistema Operacional?

Seu projeto deve focar na eficiência e na conveniência em relação ao uso do hardware.

3) Quais as características dos seguintes tipos de sistemas operacionais:

a) Lotes	c) Tempo Compartilhado	e) Paralelos
b) Batch Multi-Programados	d) Tempo Real	f) Distribuídos

Lotes: ausência de SO. O programador, além de escrever os programas, precisava escrever as tarefas de gerência do hardware utilizando cartões perfurados.

Batch: SO simples já com tarefas de gerência de processos e gerência de memória. Já era possível armazenar um pool de Jobs na memória e escolher dentre eles qual seria levado a CPU.

Tempo compartilhado: inclusão de tarefas de gerência de segurança, comunicação do usuário com o SO e inclusão do sistema de arquivos no SO. Vários usuários utilizando terminais simples compartilhando o mesmo hardware.

Tempo real: lidam com interações on-line do SO para com o ambiente no qual ele atua, estes sistemas devem implementar estratégias de tratamento de erros e tolerância a falhas.

Paralelos: SO's para máquinas com mais de uma CPU dentro da mesma placa mãe, devem implementar estratégias para gerência da comunicação entre os processos que estão em execução em diferentes CPU's. Sistemas fortemente acoplados os processos compartilham os recursos de hardware existentes na máquina.

Distribuídos: tratam problemas similares aos sistemas paralelos só que utilizando vários computadores independentes em comunicação ativa, através da rede de computadores. Os SO's agora precisam lidar com tarefas envolvendo a segurança e o tráfego de informações através da rede de computadores. Sistemas fracamente acoplados a parada de uma máquina não implica na parada do sistema.

4) Descreva as seguintes arquiteturas de Sistemas Operacionais:

a. Monolíticos	c. Máquinas virtuais
b. Em camadas	d. Microkernels

Monolíticos: SO implementado como uma série de procedimentos e funções que se comunicam através da passagem de parâmetros. Implementação confusa e de difícil manutenção.

Em camadas: O SO é implementado como uma série de camadas, cada camada se comunica somente com a camada imediatamente superior e inferior. Níveis de camadas: drivers, kernel, subsistemas do SO e a API.

Máquinas virtuais: simulação de hardware via software, causam a impressão da existência de várias máquinas dentro de um mesmo hardware. Permitem a execução de sistemas operacionais independentes dentro de cada máquina virtual.

Microkernels: criação de um kernel pequeno eficiente e estável. Toda a parte visual e de comodidade do SO é implementada de forma independente do kernel, alterações nesta camada não implicam em alterações no kernel.

- 5) O que são chamadas ao sistema (System Calls)?
São o mecanismo através do qual as aplicações se comunicam com o SO, funcionam como chamadas de procedimentos/funções internas do SO.
- 6) O que é um processo?
É um programa em execução, na memória RAM sob a supervisão do SO.
- 7) O que é uma thread?
É um processo leve ou uma unidade básica de utilização da CPU. Uma cópia do código fonte com referências próprias para os recursos da CPU.
- 8) Descreva o funcionamento de sistemas multi-thread:
Um *thread* compartilha com outros *threads* do mesmo processo: o código fonte, os dados/arquivos e os recursos do SO. Várias threads podem estar executando paralelamente, a comunicação entre as mesmas é feita através de variáveis compartilhadas localizadas na memória RAM.
- 9) Cite e explique os estados de um processo e as transições de estado pelas quais um processo pode passar:
Novo: O processo está sendo criado
Pronto: O processo está esperando para ser atribuído a um processador
Em execução: Instruções estão sendo executadas
Em espera: O processo espera por um evento
Encerrado: O processo terminou sua execução
Pronto <-> Execução (preempção)
Execução -> Espera (programa a espera de um evento)
Espera -> Pronto (evento detectado pelo SO)
- 10) Explique os critérios de otimização, que devem ser considerados ao se escolher ou projetar um algoritmo de escalonamento de processos.
Utilização de CPU: manter a CPU o mais ocupada possível.
Throughput: quantidade de processos que completam sua execução por unidade de tempo.
Tempo de retorno: quantidade de tempo para executar um processo.
Tempo de espera: quantidade de tempo que um processo gasta na fila de processos prontos.
Tempo de resposta: quantidade de tempo gasto entre a requisição e a produção da primeira resposta.
- 11) O que é escalonamento de processos preemptivo e não-preemptivo?
Escalonamento *não-preemptivo* um processo só perde a CPU por vontade própria.
Escalonamento *preemptivo* o processo em execução pode perder a CPU para outro e maior prioridade.
- 12) Com base nos dados abaixo mostre os diagramas de escalonamento para os algoritmos: FIFO, SJF Preemptivo, SJF Não-Preemptivo, por Prioridade Preemptivo, por Prioridade Não-Preemptivo e Round-Robin (quantum = 2).

Prioridade	Processo	Chegada	Tempo Surto												
3	P1	0.0	6												
2	P2	2.0	2												
1	P3	4.0	4												
4	P4	5.0	2												
FIFO	1	1	1	1	1	1	2	2	3	3	3	3	4	4	
SJFNP	1	1	1	1	1	1	2	2	4	4	3	3	3	3	
SJFP	1	1	2	2	1	4	4	1	1	1	3	3	3	3	
PNP	1	1	1	1	1	1	3	3	3	3	2	2	4	4	
PP	1	1	2	2	3	3	3	3	1	1	1	1	4	4	
RR	1	1	2	2	1	1	3	3	4	4	1	1	3	3	

- 13) Em um sistema operacional o escalonador de processos utiliza duas filas, a fila “A” contém os processos do pessoal do CPD e a fila “B” contém os processos dos alunos. O algoritmo entre filas é o Round Robin onde: de cada 11 unidades de tempo (quantum), 7 são fornecidas para os processos da fila “A” e 4 para os processos da fila “B”. O algoritmo interno de cada fila também é o Round Robin, com quantum igual a 2 unidades de tempo. A tabela abaixo mostra o conteúdo das duas filas no instante zero. Considere que esta iniciando um ciclo de 11 unidades e agora a fila “A” vai receber as suas 7 unidades de tempo. Mostre a seqüência de execução dos processos:

Obs: as filas “A” e “B” estão ordenadas pelos nomes dos processos. Se terminar a fatia de tempo da fila “X” no meio da fatia de tempo de um dos processos, o processador passa para a outra fila. Entretanto, esse processo permanece com o primeiro da fila “X”.

Fila	Processo	Duração
A	P1	6
A	P2	5
A	P3	7
B	P4	3
B	P5	8
B	P6	4

RESTA	4	3	5	3	1	6	2	1	3	0
EXECUÇÃO	11	22	33	1	44	55	1	22	33	11
FILA-PROX	A – P1(1)					B – P6(2)		A – P2(1)		

RESTA	2	0	5	0	0	4	0	3	0
EXECUÇÃO	66	4	5	2	333	5	66	5	555
FILA	B – P5(1)			A - FIM		B – P5(1)		B - FIM	

- 14) Proponha uma solução de escalonamento de processos, através de filas múltiplas com ou sem realimentação, para o seguinte problema de uma empresa, existem:

- Processos lentos, acima de 3hs/processo, sem restrição de data e hora para executarem.
- Processos de alta prioridade para a empresa.
- Processos dos usuários de tempo variável, sendo interessante executar os orientados a E/S primeiro.

FILA	PROCESSOS	ALGORITMO DA FILA
A	ALTA PRIORIDA	RR(20)
B1	USUÁRIO – E/S	RR(20)
B2	USUÁRIO – CPU	RR(20)
C	LENTOS	FIFO

O ALGORITMO ENTRE FILAS É O RR SENDO (50% - A) (30% - B1) (15% - B2) (5% - C)
SE UM PROCESSO EXECUTAR UMA FATIA DE QUANTUM NA FILA B1 ELE PASSA PARA A FILA B2, DE MENOR PRIORIDADE.