

SISTEMAS OPERACIONAIS I

LISTA 1

1) Quais as duas principais funções de um Sistema Operacional?

- Prover uma abstração da máquina, de forma a simplificar detalhes de operação do hardware e facilitar sua programação/utilização. (SO como uma máquina estendida)
- Gerenciar recursos de hardware da máquina, tanto no tempo (Ex. CPU entre usuários) quanto no espaço (Ex. Múltiplos arquivos em um disco rígido). (SO como gerenciador de recursos)

Gerenciar Recursos – O sistema operacional deve gerenciar a utilização dos recursos fornecidos pelo hardware, como processadores, memória, dispositivos de E/S, de modo que mantenha o controle sobre qual usuário/programa utiliza qual recurso, compartilhando os recursos entre os usuários/programas de modo seguro e sem conflitos.

Estender a máquina - O sistema operacional deve oferecer ao usuário uma maneira mais acessível de programar/utilizar o hardware do que as próprias instruções que este oferece. Por exemplo: o usuário não precisa saber qual a trilha e o setor do disco se deseja gravar alguma informação, apenas faz uma chamada ao sistema que estende as instruções de E/S, disponibilizando instruções mais amigáveis para estas e outras tarefas. Ou seja, o sistema operacional atua como uma interface entre o hardware e o ambiente de software.

2) O que é multiprogramação? Cite duas razões para se ter multiprogramação?

É a técnica para desenvolvimento e execução de mais de um programa de computador em uma mesma máquina, simultaneamente. Algumas razões para sua utilização podem ser:

- Melhor utilização de recursos: se determinado programa em execução necessita fazer um acesso a disco (muito mais lento que a CPU), ao invés da CPU ficar ociosa aguardando o dado do disco estar disponível, ela pode ser alocada para processar outro programa em execução, retornando em seguida para o primeiro. Assim a máquina estará sempre sendo utilizada, evitando períodos de ociosidade.
- Operação interativa com o usuário: múltiplos usuários podem se conectar simultaneamente à máquina e despachar comandos, com a CPU dividindo seu poder de processamento entre eles. Assim, um usuário pode detectar uma falha (como um erro de digitação de um comando) em tempo de execução, sem precisar esperar que seu programa entre em uma fila de processamento e só após sua saída, identifique que existia um erro.

Multiprogramação é uma técnica utilizada em sistemas operacionais para maximizar o uso da CPU. Nesta técnica, a memória é dividida em várias partes que contém um job cada. Enquanto um job está fazendo uma operação de E/S, a CPU passa a executar outro job que esteja em outra parte da memória, evitando a ociosidade da CPU. Uma razão para se ter multiprogramação é a maximização do uso da CPU obtida por esta técnica. Outra razão é o uso mais inteligente dos recursos de hardware (CPU, memória, E/S) que na época em que a técnica surgiu eram caros e não podiam ser desperdiçados. Uma outra razão é o fato de aumentar o número de jobs concluídos por unidade de tempo, melhorando a performance do sistema como um todo.

3) O que é Spooling?

Consiste na técnica de colocar uma dada informação em um local temporário, para que outra entidade o acesse. É útil pois permite que duas entidades acessem os independentemente, sem comunicação direta

e em taxas distintas.

No caso dos computadores de segunda geração, um exemplo era o fato dos programas serem previamente escritos em fita magnética por uma máquina e posteriormente lidos e processados por uma segunda máquina, sendo em seguida impressos por uma impressora. Neste caso, os três dispositivos possuíam velocidades de operação distintas, mas o processo de Spooling permitia uma troca de informações indireta e “isolada” entre si, fazendo por exemplo, com que a máquina intermediária operasse na fila já existente, em uma velocidade mais rápida, sem precisar esperar pelas etapas mais lentas das demais máquinas.

- Simultaneous peripheral operation on-line surgiu como opção para um melhor aproveitamento da UCP e dos dispositivos periféricos (final dos anos 50).
- Utilizado, principalmente, na impressão.
- Ao invés de enviar os jobs diretamente para a impressora e ficar esperando o término da impressão para iniciar a execução de outro programa, a UCP envia o job para um arquivo intermediário (no início era uma fita). Este arquivo é, então, impresso, sendo que a ordem de impressão dos jobs é determinada por prioridades atribuídas à eles.

É uma técnica que permite que os jobs fiquem armazenados em um buffer (no disco magnético) para uso do CPU. Quando um job que está na memória termina de executar, um outro é lido deste buffer e colocado na partição onde este (o que acabou) estava. Esta técnica automatizou parte do processo de entrar com programas a serem executados (antes eram fitas que eram gravadas e levadas de um equipamento para outro). A técnica de Spooling (simultaneous peripheral operations on-line) é muito utilizada em impressão, onde há um buffer (fila) onde os arquivos a serem impressos ficam armazenados e a impressora os lê sequencialmente, liberando o PC para outro tipo de uso.

4) Defina as propriedades essenciais dos seguintes tipos de Sistemas Operacionais:

a. Batch

- Junta vários trabalhos em um lote (batch) e os executa um a um;
- Voltado à aplicações científicas;
- Operação via cartões perfurados;

Ex. Mainframes IBM (1955 - 1965) / SO Fortran Monitor System

- Jobs (tarefas), sem interação com o usuário, executados sequencialmente.
- Sort, compilações, backup, etc.
- Podem apresentar tempos de resposta longos (seqüenciais).

- Execução contínua até o fim do processamento de um lote, ou até o uso de E/S;
- Uso para aplicações CPU-Bound;
- Maximizar o uso da CPU é o fator fundamental;

b. Time-Sharing

A CPU “divide” seu poder de processamento entre múltiplos programas, executados simultaneamente. A CPU opera em um determinado programa por uma quantidade X de tempo e sem seguida o coloca em espera, passando a processar outro programa, e assim por diante, até que complete todos os programas em execução e repita a tarefa ciclicamente.

Isto permite que:

- Múltiplos usuários utilizem a máquina simultaneamente, cada um deles tendo a ilusão de ter o computador todo o tempo e de maneira interativa.

- Respostas do SO em poucos segundos.

- Para cada usuário uma fatia de tempo (time-slice) é alocada pelo processador. Caso a execução exceda a fatia de tempo ele é substituído por outro e fica esperando nova oportunidade.
- Todos os recursos são compartilhados, porém o usuário tem a impressão de estar utilizando a máquina sozinho.
- Implementação complexa porém aumentam a produtividade dos usuários, reduzindo custos de utilização.
- Compartilhamento da CPU entre os jobs (processos) ao longo do tempo;
- Uso em sistemas multiusuários/multiprocessos;
- Escalonamento de recursos;

c. Tempo-real

SOs de tempo real possuem suas tarefas classificadas por prioridade e um tempo máximo que cada uma tem para executar. Assim ele faz o gerenciamento de recursos de forma a priorizar as mais importantes e garantir que elas completem sua execução antes do tempo máximo estipulado. Não se importam em compartilhamento justo dos recursos entre os processos, sendo que a prioridade sempre será dada a tarefa de maior importância.

- Frequentemente usados como dispositivos de controle em aplicação dedicada: controle de experimentos científicos, imagens médicas, sistemas de controle industrial, etc.
- Sistemas de tempo-real podem ser hard ou soft .
- Tempo é o fator fundamental;
- Prazos rígidos para o processamento das tarefas, nem antes, nem depois (críticos) ou com uma certa folga (não crítico).
- Controle de máquinas, áudio digital, multimídia.

d. Distribuído

São aqueles que gerenciam recursos de múltiplas máquinas, interconectadas por uma rede de comunicação. Assim a memória total do SO, ou o espaço de armazenamento pode estar distribuído entre máquinas distintas, ou mais de uma CPU pode contribuir para o processamento de um programa. Os componentes do computador (CPU, memória, etc) não precisam necessariamente estar na mesma peça de hardware.

- loosely coupled system – cada processador tem sua própria memória local; processadores comunicam-se através de várias linhas de comunicação, tais como barramentos de alta velocidade e linhas telefônicas.
- Distribui a computação entre vários processadores físicos;
- Compartilhamento de recursos;

5) Descreva as diferenças entre multiprocessamento simétrico e assimétrico. Quais as vantagens e desvantagens dos sistemas multiprocessador?

Simétrico

- Cada processador executa uma cópia idêntica do SO.
- Muitos processos podem executar de uma vez sem deterioração do desempenho.
- É suportado pela maioria dos SOs modernos.

Assimétrico

- A cada processador é atribuída uma tarefa específica.
- Processador mestre escalona e aloca trabalho para os processos escravos.

- Mais comum em sistemas muito grandes.

Vantagens:

- Aumento de desempenho
- São econômicos
- Aumento da confiabilidade
- Degradação controlada(fail-soft systems)

Desvantagens:

- Problemas de sincronização
- Controle de concorrência
- Consistência das informações

Multiprocessamento assimétrico é uma técnica para sistemas multiprocessados, onde somente um processador tem acesso ao núcleo (e suas estruturas) e realiza o papel de escalonador dos outros processadores, formando uma estrutura mestre-escravo. Já no multiprocessamento simétrico, cada processador executa uma cópia idêntica do SO e muitos processos podem executar de uma vez sem deterioração do desempenho, porém surgem problemas de sincronismo entre cada cópia do SO de cada processador.

A grande vantagem dos sistemas multiprocessador é o ganho de desempenho, ao processar mais de uma tarefa ao mesmo tempo, além da confiabilidade e economia. Suas desvantagens são os inconvenientes na gerência da execução paralela.

6) Porque sistemas distribuídos são desejáveis?

Sistemas distribuídos são desejáveis porque permitem o compartilhamento de recursos, aumentam a velocidade de computação(load sharing), são confiáveis e permitem comunicações através de várias linhas, como barramentos de alta velocidade e linhas telefônicas.

Para solucionar complexos problemas computacionais, no entanto atualmente vem sendo largamente utilizado com outras finalidades, como em um sistema distribuído conectando vários computadores pela internet o qual está tipicamente envolvido com comunicação de dados.

7) Qual é a diferença entre um trap e uma interrupção? Qual é o uso de cada uma das funções?

A interrupção está relacionada com os dispositivos de hardware, enquanto a instrução trap está relacionada com chamadas de sistema de programas de usuário. Ao contrário do trap, a interrupção não é visível para o programa de usuário.

Ambas são exceções, ou seja, ambas desviam a CPU do fluxo de execução atual.

A principal diferença é que as interrupções caracterizam eventos assíncronos gerados normalmente por algum dispositivo de hardware (I/O, relógio...) que interrompe a atividade atual da CPU para que este evento seja tratado, isto é feito normalmente, através de rotinas pré-definidas no sistema operacional. Já a trap é gerada por software e indica algum evento de erro ou exceção, é normalmente utilizada para mudar o modo de execução do modo usuário para o modo kernel.

8) Para que tipos de operações o DMA é útil? Quais as implicações de uma organização sem DMA na multiprogramação?

A técnica de DMA (Direct Memory Access) é útil em operações de E/S. O controlador recebe informações de onde o dado se encontra, qual o dispositivo de E/S envolvido, posição inicial de memória onde os dados serão lidos ou gravados e o tamanho do bloco de dados. A transferência é realizada pelo controlador (que assume temporariamente o controle do barramento), utilizando uma área de memória reservada, denominada buffer.

Na multiprogramação, a técnica de DMA é útil pois permite que a CPU se dedique a atender/processar outra tarefa em execução, enquanto a operação de E/S ocorre em paralelo, sem necessitar ficar ociosa esperando a E/S.

O DMA é útil para operações em que haja a necessidade da transferência de informação em blocos de algum dispositivo para a memória principal, com o intuito do uso posterior dos dados pela CPU. O DMA permite que os dados sejam transferidos sem a necessidade do gasto de computação durante a transferência, liberando a CPU para outro processamento durante este período. Após a conclusão da transferência, o chip DMA manda um sinal de interrupção para a CPU, informando-a do fim.

Uma organização sem DMA na multiprogramação, torna mais lenta a transferência de dados presente em uma operação de E/S, pois sempre que houver a transferência de dados entre a memória e o dispositivo, haverá utilização da CPU; enquanto que através do DMA, a configuração da transferência de vários blocos é feita somente no início da transação.

9) Quais das seguintes instruções deveriam ser privilegiados?

- a. Set valor do relógio**
- b. Ler o relógio**
- c. Limpar memória**
- d. Desligar interrupções**
- e. Chavear modo usuário/sistema**

Privilegiados:

A - Set relógio

C - Limpar memória (? - se não um prog pode apagar outro q esteja na memória?)

D - Desligar interrupções

E - Chavear modo usuário/sistema (? - se não qqr programa pega root...)

10) Que características de hardware (máquina) são necessárias para suportar proteção para o kernel do SO ? Explique brevemente cada característica e como elas permitem o kernel controlar os programas de usuário.

- Timer: evita o monopólio do processador;

- Registradores base e limite: proteção de memória

Proteção de memória – não deixa uma área de memória ser afetada por programas de outra área de memória. O kernel pode alocar os processos do usuário cada um com sua área de memória, não permitindo a invasão de um processo à área do outro ou à área do sistema. É feita através de registradores base e limite e do mapeamento de endereços físicos para endereços virtuais através da MMU.

Modos de operação – Restringe algumas instruções da CPU ao sistema. A operação em modo kernel tem acesso a todas as instruções do hardware, enquanto que em modo usuário, boa parte das instruções, principalmente de E/S e configuração, não estão disponíveis. Permite que somente o kernel tenha controle sobre os dispositivos de E/S, bem como a outras configurações.

Timer e Interrupções – Não deixa que um único job tenha monopólio sobre o sistema. Permite a execução

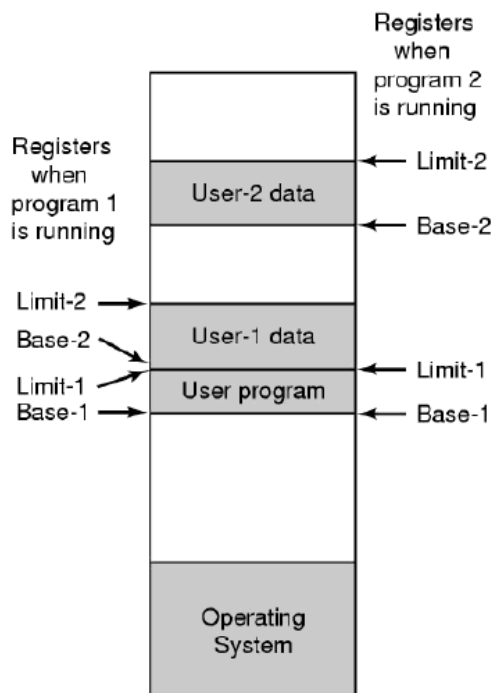
de operações nos dispositivos de E/S em paralelo com o processamento de outro job na CPU.

11) Qual é o propósito das chamadas de sistema?

As chamadas de sistema são uma interface entre o programa que está executando e o sistema operacional (funções do SO disponíveis aos programas). Tem a ver com controle de processos e manipulação de arquivos.

Permitir a programas aplicativos o acesso a recursos e operações do sistema, dentre os quais: operações de entrada e saída, comunicação inter-processos, leitura do relógio interno, manipulação de arquivos em disco, serviços de impressão, etc.

12) Usando registrador de base e registrador limite, faça um desenho da unidade de gerencia de memória (MMU) que possa prover proteção de memória entre processos?



13) O modelo cliente-servidor é popular em sistemas distribuídos. Ele pode ser usado em um sistema single-computer?

Sim, em um sistema single-computer múltiplos processos em execução podem se comunicar entre si, no mesmo paradigma que o modelo cliente - servidor. Isto contribui para a modularidade do sistema e sua estabilidade. Exemplo: Unix - interface gráfica opera como um módulo cliente, podendo estar na mesma máquina ou não.

Pode, pois este modelo apenas especifica a estrutura do sistema através de processos clientes e processos servidores (que detêm as funcionalidades do sistema) e ambos são executados em modo usuário. Nesse modelo, o sistema apenas trataria da comunicação entre estes processos e das políticas envolvidas nessa comunicação. Portanto, essa estrutura pode ser implementada em modo local, e em sistemas distribuídos, apenas diferenciando-se o modo de comunicação entre os processos.

Lista 2 – Programação Concorrente e Gerência de Processos

1) O sistema operacional é um programa dirigido por eventos, onde eventos são sinalizados por interrupções. Para cada um dos 3 principais tipos de interrupções (periférico, proteção, chamada de sistema) descreva a reação que o sistema operacional deverá ter. Em outras palavras, o que o sistema operacional deverá fazer em função do evento sinalizado.

2) Os três principais estados de um processo são:

- Esperando pelo processador (ready);
- Executando (running);
- Esperando pela entrada/saída (blocked).

Descreva os eventos que fazem com que um processo mude de estado.

R: O "execução" muda para "bloqueado" se necessitar de uma entrada, e muda para pronto caso seu tempo acabe e o escalonador selecione outro processo.

O bloqueado muda para pronto caso a entrada esteja disponível.

O pronto muda para execução se o escalonador o selecionar para tal.

3) A operação "passa para modo usuário" deve ou não ser privilegiada ? Justifique.

4) A operação "Desabilita interrupções" deve ou não ser privilegiada ? Justifique.

5) A operação "escreve caracter na interface da impressora" deve ou não ser privilegiada? Justifique.

6) A operação "desliga o timer" deve ou não ser privilegiada ? Justifique.

7) Pode-se considerar como consequência da multiprogramação "uma pior utilização da cpu" ? Justifique.

R: Não, pelo contrário, a CPU é melhor utilizada pois quando o programa faz uma requisição de entrada de um dado, por exemplo, a CPU não fica ociosa e é escalonado outro job para ela enquanto o dado não é carregado.

8) Pode-se considerar como consequência da multiprogramação "uma pior utilização dos periféricos" ? Justifique.

R: Não, pois o sistema pode buscar dados de um periférico mesmo com a utilização da cpu por outro processo.

9) Pode-se considerar como consequência da multiprogramação "uma menor necessidade de memória" ? Justifique.

R: Não, pois em consequência da multiprogramação é necessário mais memória para armazenar todos os programas em execução.

10) Pode-se considerar como consequência da multiprogramação "uma menor necessidade de hardware para proteção" ? Justifique.

R: Não, em consequência da multiprogramação devemos ter uma maior hardware para proteção, para evitar que um programa invada a área de memória de outro, etc.

11) Muitas arquiteturas dividem as instruções em normais e privilegiadas (restritas). Mostre como isto pode ser utilizado para impedir que processos tenham acesso direto aos periféricos, mas ainda possam fazer as operações de entrada e saída necessárias (descreva o mecanismo).

12) Explique quando são necessários e quais as funções dos escalonadores de curto-prazo, médio-prazo e longo-prazo.

13) Crie um algoritmo de escalonamento baseado em filas multinível com realimentação. Devem existir duas filas. O algoritmo entre filas deve trabalhar de forma que, com o passar do tempo, processos i/o-bound vão para a fila 1, e processos cpu-bound para a fila 2. Não deve ser possível a ocorrência de postergação indefinida de nenhum processo.

14) Que tipos de critérios devem ser utilizados no momento da definição da fatia de tempo a ser empregada em um determinado sistema?

15) Em um sistema operacional, o escalonador de curto prazo está organizado como 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 round-robin. De cada 11 unidades de tempo de cpu, 7 são fornecidas para os processos da fila A e 4 para os processos da fila B. O tempo de cada fila é dividido entre os processos também por round-robin, com fatias de tempo de 2 unidades para todos. A tabela abaixo mostra o conteúdo das duas filas no instante zero. Considere que está iniciando um ciclo de 11 unidades, e agora a fila A vai receber as suas 7 unidades de tempo. Mostre a sequência de execução dos processos, com os momentos em que é feita a troca (diagrama de Gantt).

OBS: Se terminar a fatia de tempo da fila X no meio da fatia de tempo de um dos processos, a cpu passa para a outra fila. Entretanto, este processo permanece como primeiro da fila X, até que toda sua fatia de tempo seja consumida.

Fila	Processo	Duração do próximo ciclo de cpu
----	-----	-----
A	P1	6
A	P2	5
A	P3	7
B	P4	3
B	P5	8
B	P6	4

16) Quatro programas devem ser executados em um computador. Todos os programas são compostos por 2 ciclos de cpu e 2 ciclos de entrada e saída. A entrada e saída de todos os programas é feita sobre a mesma unidade de disco. Os tempos para cada ciclo de cada programa são mostrados abaixo:

<u>Programa</u>	<u>Cpu</u>	<u>Disco</u>	<u>Cpu</u>	<u>Disco</u>
P1	3	10	3	12
P2	4	12	6	8
P3	7	8	8	10
P4	6	14	2	10

Construa um diagrama de tempo mostrando qual programa está ocupando a cpu e o disco a cada momento, até que os 4 programas terminem. Suponha que o algoritmo de escalonamento utilizado seja round-robin, com fatias de tempo de 4 unidades. Qual a taxa de ocupação da cpu e do disco?

17) O que acontece com as duas taxas de ocupação calculadas no problema anterior se for utilizado um disco com o dobro da velocidade de acesso (duração dos ciclos de e/s é dividida por 2)?

QUIZ Exercício de escalonamento

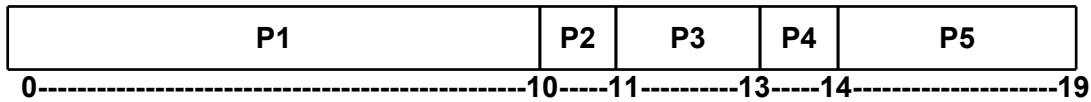
Considere o seguinte conjunto de processos, com o tamanho do ciclo de UCP (CPU-burst time) dado em milissegundos:

Process	Ciclo de UCP	Prioridade
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

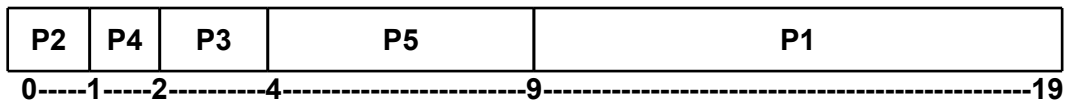
Assumir que os processos chegam na ordem P1 , P2 , P3 , P4 , P5 , todos no tempo 0.

a. Desenhe 4 diagramas de Gantt para mostrar a execução destes processos utilizando as seguintes políticas de escalonamento: FCFS, SJF, Prioridade (numero menor implica em prioridade maior) e RR com quantum = 1.

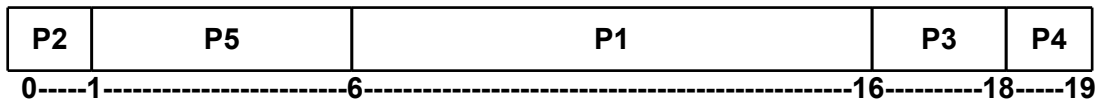
FCFS



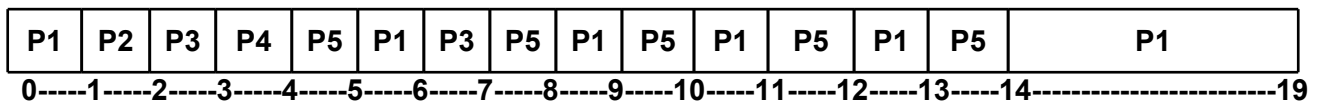
SJF



PRIO



RR



b. Qual é o tempo de turnaround de cada processo em cada algoritmo de escalonamento acima? Qual é o valor médio?

Tempo de turnaround = tempo de espera + tempo de execução

	FCFS	SJF	PRIO	RR
P1	10	19	16	19
P2	11	1	1	2
P3	13	4	18	7
P4	14	2	19	4
P5	19	9	6	14
Média	13,4	7	12	9,2

c. Qual é o tempo de espera de cada processo em cada algoritmo de escalonamento acima? Qual é o valor médio?

Tempo de espera = tempo de turnaround – tempo de ciclo

	FCFS	SJF	PRIO	RR
P1	0	9	6	9
P2	10	0	0	1

P3	11	2	16	5
P4	13	1	18	3
P5	14	4	1	9
Média	9,6	3,2	8,2	5,4

d. Qual dos escalonamentos resulta no mínimo tempo médio de espera?

R: O SJF (Short Job First)

Conteúdo provável de cair:

Estrutura do MINIX

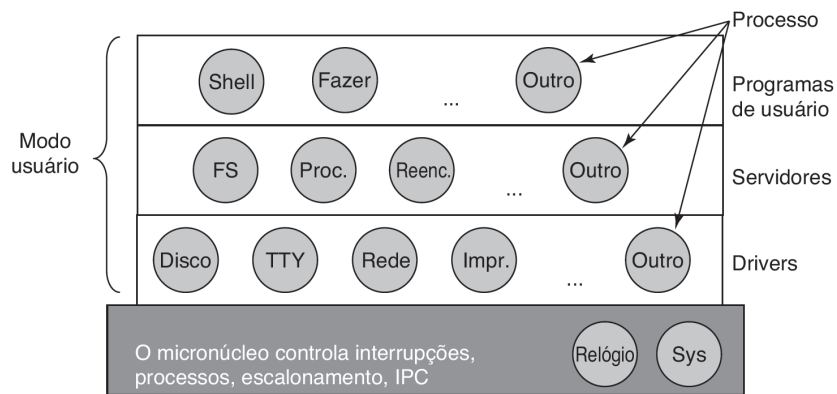


Figura 1.23 Estrutura do sistema MINIX 3.

Não sei se essa é a figura certa/melhor para descrever a estrutura no MINIX (que comentaram que deve cair)

vo começa pela mais estressante, a questao do fork.. era uma questao que dava o programinha com 4 fork()'s e tinha que montar a arvore de processos (era a ultima questao da prova)..

o codigo era esse:

```

1:   c2 = 0;
2:   c1 = fork();
3:   if (c1 == 0)
4:       fork();
5:   c2 = fork();
6:   if (c2 > 0)
7:       fork();

```

A primeira pergunta é "qtos processos sao criados" (acho que a repsosta é 8, errei na hoa e corrigi depois em casa ⇐) e a segunda é pra gerar a arvore de processos (comecando do 0

e aumentando o numero a cada processo novo, vo te passa um desenho em anexo, que pela minha correcao depois ta certa lol). Hint: Na hora de resolver atencao nos valores de c1 e c2 em cada processo.

Outra questao de exercicio era pra fazer a sequencia do algoritmo (diagrama de tempo, acho que era grant o nome, algo assim) de escalonamento RR (Round Robin) em 4 processos que ele dava. (o quantum era 2).

processo:	tempo de chegada:	uso do CPU:
A	0	10
B	3	6
C	5	6
D	7	4

dai ele dizia que a regra era que cada processo que usasse seu quantum ia pro final da fila, e a cada processo novo que chegasse ele vinha pro inicio da fila.

entao, se do jeito que eu fiz ta certo.. o processo A usa 2q, nao chegou ninguem entao ele faz mais 2q, chegou B no tempo 3 entao depois o B faz 2q, (chegou C) o C faz 2q, (chegou D) o D usa 2q, depois A2q, B2q, C2q, D2q, A2q, B2q, C2q, (D ja acabou) A2q, B2q, C2q, A2q, (B e C acabaram)

A2q.

Entao acho que ficou algo como A,A,B,C,D,A,B,C,D,A,B,C,A.

Dai perguntava o tempo medio de espera e o turn around medio.

Das perguntas discursivas lembro poucas.. mas tinha uma que perguntava "Descreva a arquitetura do MINIX".

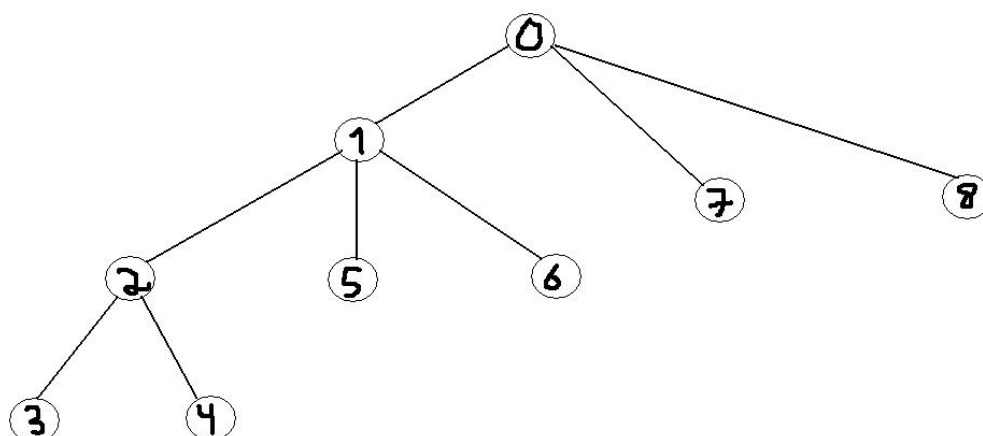
Tinha 2 questoes de verdadeiro e falso que em geral perguntavam coisas como "tal coisa significa isso..".

e lembrando algumas coisas q vao cair

- arvore de processos,

- camadas minix

round robin



Tópicos interessantes para a P1:

O que é make e makefile

- make lê um arquivo texto e makefile executa suas operações de acordo com regras de dependência

transição do modo usuário para o modo kernel

- processo de usuário executando uma tarefa
- chamada de sistema (usuário)
- trap mode bit = 0
- executa a chamada de sistema (kernel)
- retorna mode bit =1
- retorna da chama de sistema (usuário)

sistemas multitarefas: preemptivo e cooperativo

- preemptivo: existe uma democratização do uso do processador
- cooperativo: apenas em partes do código o CPU está liberado para ser dividido com outros processos

chamadas de sistema

interface de programação para serviços do SO

implementação de uma chamada

tipicamente é um número associado a uma tabela

o que é máquina virtual

é uma maneira de tratar o kernel do SO e o hardware como se fossem tudo hardware, forma de fazer com que o sistema operacional tivesse seu próprio CPU

o que é microkernel

- termo usado para indicar que as funcionalidades do SO saíram do kernel e foram para servidores, assim usando o mínimo possível de espaço do sistema, assim deixando o máximo de espaço para o modo usuário
- o microkernel é responsável pelo tratamento de interrupções e exceções, escalonamento de processos, contador de instruções de programa ICP

estrutura do sistema microkernel

- modo usuário
- modo kernel
- servidores