

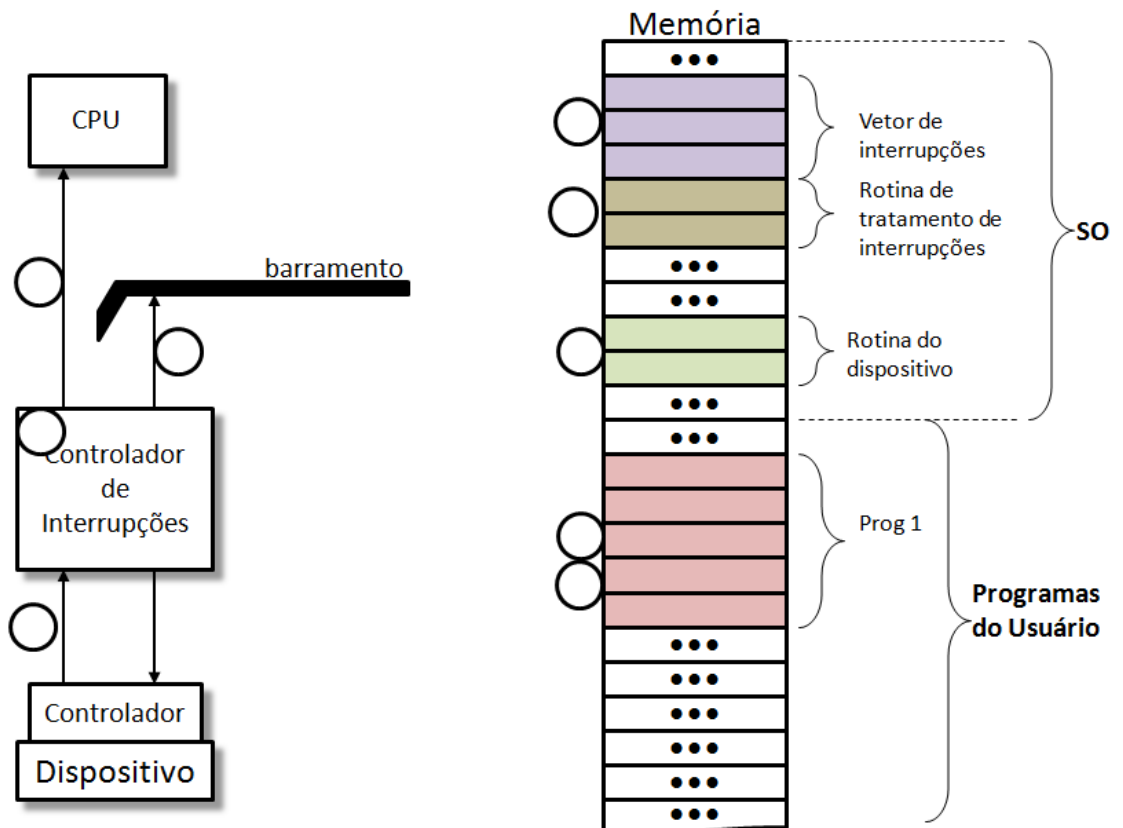
Exercícios Propostos

- (1) O que são e para que servem Sistemas Operacionais?
- (2) Na pilha de abstrações que utilizamos em GSI013 e agora em GSI018, em que lugar se encaixam os sistemas operacionais? Discuta como ocorre o interfaceamento entre as camadas imediatamente superior e a inferior?
- (3) Explique porque computadores de propósito específico como o colossus ou o ENIAC não requeriam sistemas operacionais para funcionarem.
- (4) Descreva em suas palavras como foi a transição de sistemas computacionais completamente sem sistemas operacionais para a criação e especialização paulatina de sistemas computacionais dependentes de sistemas operacionais. (Lembrem-se das bibliotecas de E/S, programa de inicialização, maximização da potencialidade do sistema computacional, etc...)
- (5) Quais foram as motivações que levaram a criação de sistemas multiusuário?
- (6) Quais foram as motivações que levaram a criação de sistemas multitarefas?
- (7) Como sistemas multitarefas e multiusuário se relacionam?
- (8) Porque sistemas operacionais eram escritos diretamente em assembly e não em uma das linguagens de alto nível disponíveis tais como FORTRAM ou COBOL?
- (9) Como a criação da linguagem C para originalmente escrever o UNIX modificou a forma como escrevemos sistemas operacionais modernos?
- (10) Descreva o funcionamento de um sistema operacional em lotes.

- (11) Em sistemas de multiprogramação podemos definir processos como CPU Bound ou I/O bound. Qual a diferença entre eles?
- (12) Em um sistema operacional moderno típico, quais são os seis serviços principais providos pelo sistema operacional?
- (13) Liste todas as abstrações que o sistema operacional cria para tornar usável um sistema computacional.
- (14) Descreva o que é a BIOS e o que geralmente está contido nela.
- (15) Para que serve o POST?
- (16) Qual a função do Gerenciador de Processos? Descreva de maneira resumida como ele funciona.
- (17) Qual a função do Gerenciador de Memória? Descreva de maneira resumida como ele funciona.
- (18) Qual a função do Gerenciador do Sistema de Arquivos? Descreva de maneira resumida como ele funciona.
- (19) Porque a abstração do conceito de Arquivo torna a utilização do sistema computacional mais simples. Exemplifique via um diagrama ou descreva textualmente.
- (20) Qual a função do Gerenciador de Entrada e Saída de Dados? Descreva de maneira resumida como ele funciona.
- (21) Qual a função do Gerenciador de Segurança e Autenticação? Descreva de maneira resumida como ele funciona.
- (22) Com relação a estrutura dos Sistemas Operacionais modernos, descreva as principais maneiras de estruturá-los.

- (23) Porque é comum escrevermos sistemas operacionais em pelo menos duas camadas, uma dependente e outra independente do hardware? Qual a vantagem decorrente desta forma de estruturar o sistema?
- (24) Explique o conceito de Máquinas Virtuais. Discorra sobre como Máquinas Virtuais eram utilizadas antigamente e cite algumas das utilizações modernas.
- (25) Qual a diferença entre hipervisores de Tipo 1 e 2. Exemplifique graficamente.
- (26) Discorra sobre o conceito de chamadas de sistema (Syscalls).
- (27) O que são interrupções e como elas funcionam?
- (28) Embora a arquitetura von Neumann seja ainda hoje o modelo utilizado para a criação de sistemas computacionais, várias melhorias e pequenas alterações foram propostas e são comumente empregadas. Discorra sobre o assunto.
- (29) O que são North e South Bridge? Qual a responsabilidade de cada uma delas e porque elas são necessárias?
- (30) Como sabemos, a hierarquia de memória tem como objetivo criar a ilusão de que a memória disponível ao sistema seja o maior possível e o mais rápida possível. Em que partes da hierarquia de memória o SO atua para o seu funcionamento?
- (31) Na arquitetura MIPS estudada em GSI013 e utilizada em GSI018 como caso de estudo, como é organizada a memória disponível? Desenhe um diagrama de memória exemplificando-a e listando seus limites de segmentos. Considere que o barramento de endereços é de 32 bits.
- (32) Considerando que a memória do sistema seja indereçável byte-a-byte, quantos endereços distintos um sistema de 64 bits seria capaz de endereçar?

- (33) Qual a diferença entre Código de Máquina e linguagem Assembly?
- (34) Porque é importante que os dispositivos de entrada e saída funcionem de maneira desacoplada do processador?
- (35) Explique em suas palavras o conceito de interrupções. Porque elas são importantes para o funcionamento do sistema e explique passo a passo como elas funcionam.
- (36) Qual a diferença entre interrupções de hardware e de software?
- (37) O Diagrama abaixo exemplifica o funcionamento de interrupções de hardware. Utilize os círculos em branco para numerar as etapas do processo. Descreva cada uma das etapas textualmente.



- (38) O que são e para que servem temporizadores?

- (39) O que significa o acrônimo RTC no contexto de sistemas computacionais e quais são suas atribuições?
- (40) Como sabemos temporizadores contam o número de tiques do cristal oscilador e geram uma interrupção quando o número pré estabelecido é alcançado. Calcule Quantos tique seriam possíveis de se armazenar em contadores de 32, 40, 60 e 64 bits.
- (41) Considerando um clock de 500MHz responda:
- (a) Qual a periodicidade do sinal gerado por este clock?
 - (b) Para gerar uma interrupção a cada 30ms quantos tiques seriam necessários?
 - (c) E caso a interrupção devesse ser gerada a cada 100ms?
 - (d) Contadores de quantos bits seriam minimamente necessários para 30 e 100ms?
- (42) Porque são necessários e quais são as vantagens e desvantagens dos temporizadores por software?
- (43) Defina com suas palavras o conceito de Processo?
- (44) Discorra sobre o interrelacionamento entre processos e multiprogramação.
- (45) Quais são as vantagens e desvantagens de se definir um quanta longo?
- (46) No contexto de gerenciamento de processos, o que vem a ser um PCB?
- (47) Quais são os principais estados em que um processo pode estar? Cite o(s) possível(is) eventos que colocam um processo em cada um destes estados.
- (48) Porque um processo pode entrar no estado ZUMBI? Qual a vantagem de se manter um PCB de um processo que não está mais apto a ser executado?
- (49) O que significa POSIX? Com que propósito ela foi criada e para que serve?

- (50) No LINUX, há em acórdância com a especificação POSIX, três formas de se criar processos. Quais são as chamadas do sistema que implementam cada uma destas formas? Qual a diferença entre elas?
- (51) O que é hierarquia de processos e porque todo processo deve ter um pai? Há alguma exceção?
- (52) Qual a função da chamada de sistema <wait>?
- (53) O que é e de que fato decorre a necessidade do escalonador de processos?
- (54) Justifique porque não há a necessidade de um escalonador de processos em sistemas monotarefas.
- (55) Porque chavear processos é caro? Descreva os passos necessários para que ocorra uma troca de processos.
- (56) Citamos em sala de aula que quanto mais rápida a CPU maior será a tendência dos processos de serem I/O Bound. Justifique com exemplos esta afirmativa.
- (57) Em sistemas iterativos, monousuário, e multitarefas tais como o windows ou LINUX indiferentemente da velocidade da CPU os processos tendem a ser I/O Bound. Justifique esta afirmativa.
- (58) Um escalonador de processos deve alocar o processador para cinco processos distintos. Não há remoção nem criação de novos processos. Considerando que o clock do sistema seja de 500MHz, que o quanta de tempo seja de 50ms e que o escalonador gaste 10ms para chavear dois processos, quanto tempo decorrerá entre o processo A terminar sua execução e voltar a ser executado? Qual a porcentagem do tempo do processador gasta pelo escalonador de processos neste contexto?
- (59) porque chavear processos é tão custoso? Liste os passos necessários para que o chaveamento de processos ocorra.

- (60) Qual a diferença entre algoritmos preemptivos e não-preemptivos de escalonamento.
- (61) Como vimos em aula, existem diversos algoritmos de escalonamento. Alguns deles primam por objetivos distintos. No entanto todos os algoritmos concordam em três objetivos. Quais são eles e de que tratam?
- (62) Descreva como o algoritmo de escalonamento FCFS funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.
- (63) Descreva como o algoritmo de escalonamento SJF funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.
- (64) O algoritmo SJF requer que o tempo de execução do processo seja conhecido a priori. Cite exemplos em que este tempo de execução pode ser conhecido. É possível estimar estes tempos em outros contextos?
- (65) Considere que um escalonador de processos que implementa o algoritmo SJF esteja em execução. Há no sistema 5 processos a serem escalonados no processador, A(5), B(15), C(3), D(10) e E(7). Em que ordem tais processos seriam escalonados? Desenhe um diagrama mostrando ao longo do tempo como estes processos seriam escalonados. Qual seria o TMR (tempo médio restante)?
- (66) Considere que um escalonador de processos que implementa o algoritmo SJF esteja em execução. Há no sistema 5 processos a serem escalonados no processador, A(5), B(15), C(3), D(10) e E(7). Os processos são criados em tempos distintos, são eles $T_A=3$, $T_B=0$, $T_C=10$, $T_D=7$, $T_E=9$. Em que ordem tais processos seriam escalonados? Desenhe um diagrama mostrando ao longo do tempo como estes processos seriam escalonados. Qual seria o TMR (tempo médio restante)?
- (67) O algoritmo de escalonamento SRTN é uma variante preemptiva do SJF. Refaça o exercício (66) considerando que o algoritmo SRTN é utilizado. Há

alguma variação no diagrama de agendamento da CPU? Há algum impacto no TMR?

(68) Descreva como o algoritmo de escalonamento Round-Robin funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.

(69) Considere um gerenciador de escalonamento que implementa o algoritmo Round-Robin. Em um dado momento os processos B, C, D, E e F estão agendados para execução e A acaba de receber o processador. O quantum do sistema está definido para 20ms e o chaveamento de processos demora 5ms. Todos os processos neste sistema são CPU-bound, e executam por todo o seu quantum.

(a) Quanto tempo decorrerá entre o término da execução de A e que ele receba novamente a CPU?

(b) Desenhe um diagrama de como fica a fila de processos entre o agendamento de A e seu reagendamento.

(c) O que acontece quando o processo G é agendado no sistema para execução. Assuma que o agendamento ocorre durante a execução de D.

(70) Embora o algoritmo Round-Robin seja simples e largamente utilizado ele, em sua forma mais simples possui uma limitação séria. Todos os processos são tratados igualmente. Variantes do RR podem incorporar uma informação adicional que codifica quão importante o processo é para o sistema em geral. Como poderíamos ajustar o algoritmo RR para levar em consideração as prioridades dos processos?

(71) Descreva como o algoritmo de escalonamento SPN funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.

(72) Uma das soluções possíveis para estimar o tempo de execução de um processo é utilizar seu histórico de alocação. Considere a seguinte equação:

$$E = a \times T_0 + (1 - a)T_1$$

- onde "a" é a taxa de envelhecimento ("a" = [0,1])
- "T₀" é o tempo de execução na alocação anterior
- "T₁" é o tempo de execução na alocação atual

- "E" é o valor estimado do tempo de execução do processo

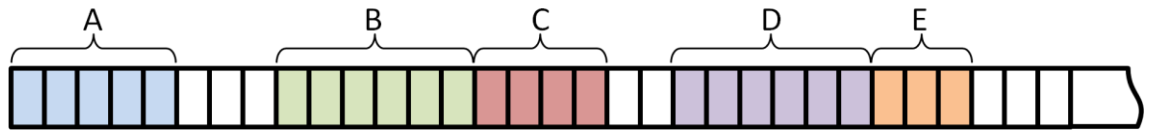
Responda:

- (a) Qual o impacto de se utilizar uma taxa de envelhecimento "a" próxima de 0 ou próxima de 1?
- (b) Suponha que um processo A qualquer tenha sido agendado seis vezes pelo processador. os tempos de CPU das seis alocações são dados por $T = \{15, 12, 22, 30, 35, 14\}$ ms. Suponha ainda que a taxa de envelhecimento "a" seja de 1/2. Qual o tempo estimado "E" no instante 2, 3, 4, 5 e 6?
- (c) Refaça o item (b) utilizando uma taxa de envelhecimento de 0.1;
- (d) Refaça o item (b) utilizando uma taxa de envelhecimento de 0.9;
- (73) Descreva como o algoritmo de escalonamento GARANTIDO funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.
- (74) Descreva como o algoritmo de escalonamento por Loteria funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.
- (75) Descreva como o algoritmo de escalonamento por Fração Justa funciona. Como ele pode ser implementado? Esboce um pseudo-algoritmo.
- (76) Quais são as três principais formas de comunicação entre processos? Detalhe-as.
- (77) Descreva em suas palavras o funcionamento de PIPES. Sinta-se a vontade para apresentar código que exemplifique seu funcionamento.
- (78) Descreva em suas palavras o funcionamento de MEMÓRIA COMPARTILHADA. Sinta-se a vontade para apresentar código que exemplifique seu funcionamento.
- (79) Descreva em suas palavras o funcionamento de SOCKETS. Sinta-se a vontade para apresentar código que exemplifique seu funcionamento.

- (80) O que é o problema da CONDIÇÃO DE CORRIDA? Cite exemplos de situações em que ele pode ocorrer?
- (81) Defina o conceito de REGIÃO CRÍTICA? Quais são as quatro condições que definem uma região crítica?
- (82) Quais são os problemas associados com o tratamento de regiões críticas por meio de desabilitação de interrupções?
- (83) O que são Locks e como eles funcionam?
- (84) Como discutimos em aula, com a evolução dos sistemas operacionais suporte em hardware para o seu funcionamento se tornou uma necessidade. Este fato ocorre em diversos contextos dentro de sistemas operacionais. Descreva o funcionamento da instrução TSL - Test and Lock.
- (85) O que são semáforos e como eles funcionam?
- (86) Defina o que são Threads? Como elas se relacionam com processos?
- (87) Monte uma tabela listando as diferenças entre Processos e Threads?
- (88) Cite e descreva alguns exemplos nos quais a utilização de Threads seja benéfica ao funcionamento de um sistema? Há casos em que Threads mais atrapalham do que auxiliam? Se sim, cite exemplos.
- (89) Descreva em suas palavras o objetivo de cada uma das seguintes funções definidas no padrão POSIX IEEE 1003.1c-2001.
- pthread_create
 - pthread_exit
 - pthread_join
 - pthread_yield
 - pthread_attr_init
 - pthread_attr_destroy

- (90) Threads podem ser implementadas no espaço do usuário e no kernel do sistema operacional. Quais são as vantagens e desvantagens de cada uma das implementações?
- (91) O Gerenciador de Memória tem como objetivo principal fornecer aos processos um espaço de endereçamento. Este espaço de endereçamento gera um conjunto de endereços virtuais que precisam de tradução para endereços físicos antes que a memória possa ser acessada. Há alguma forma de implementar multiprogramação sem a utilização de endereços virtuais? Se sim, descreva o funcionamento.
- (92) O que é realocação estática e como ela funciona?
- (93) Antes do advento da memória virtual, alguns sistemas operacionais implementavam multiprogramação. Para tal eles utilizavam hardware de proteção. Descreva como este esquema funcionava.
- (94) Descreva o conceito de espaço de endereçamento.
- (95) Como funciona o esquema de realocação intitulado REGISTRADORES BASE|LIMITE.
- (96) Em relação as políticas de alocação de memória estudamos duas formas principais. São elas Mapas de Bits e Listas Livres. Descreva em suas palavras como elas funcionam.
- (97) Considerando o esquema de alocação de memória intitulado Mapa de Bits, considere um sistema que utiliza blocos de 32 bits. Com uma memória RAM de 8GB qual seria o tamanho do mapa de bits em bytes?
- (98) Considerando o esquema de alocação de memória intitulado Mapa de Bits, considere um sistema que utiliza blocos de 2KBytes. Com uma memória RAM de 8GB qual seria o tamanho do mapa de bits em bytes?
- (99) Considerando o esquema de alocação de memória intitulado Listas Livres. A memória poderia estar

alocada como o diagrama abaixo. Desenhe a lista encadeada que representa este esquema de alocação.



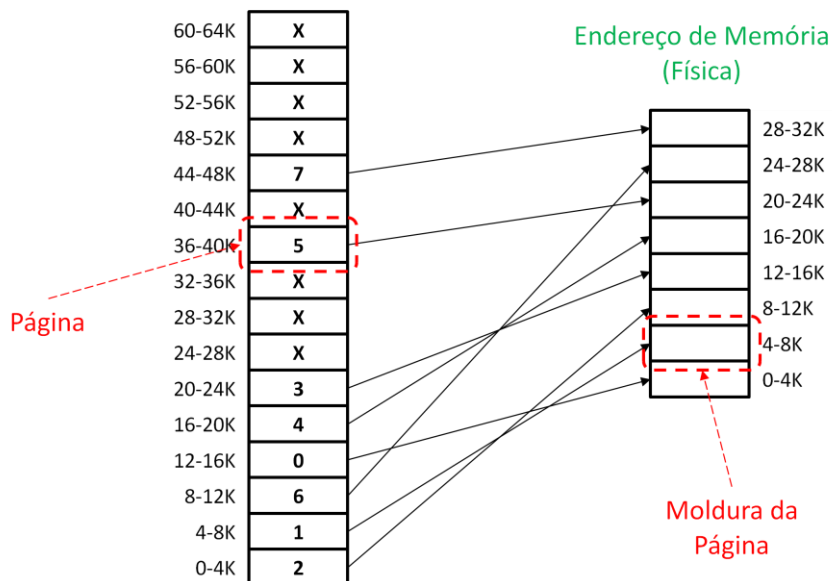
(100) Com relação aos algoritmos de Alocação de memória. Descreva como os seguintes algoritmos funcionam e quais as suas vantagens e desvantagens:

- (a) FIRST FIT
- (b) NEXT FIT
- (c) BEST FIT
- (d) WORST FIT
- (e) QUICK FIT

(101) O que é memória virtual e porque ela é necessária em sistemas operacionais modernos?

(102) Para a implementação de um esquema de memória virtual faz-se necessário algum suporte em hardware. Um módulo adicional entitulado MMU é adicionado ao processador. Descreva de que trata a MMU e quais as suas atribuições.

(103) Para o exercício a seguir considere o seguinte esquema:



(a) a CPU deseja executar a instrução `sw $s3, 4($s0)`. Considere que o registrador `s3` contém o número 42 e o

registrador s0 contenha o endereço 0x0500. Qual o endereço virtual enviado a MMU e qual o endereço físico enviado pela MMU ao gerenciador de memória.

(b) O que ocorre quando tentamos acessar por exemplo uma posição de memória localizada na faixa 12-16k?

(104) Monte um diagrama mostrando o funcionamento da MMU para a tradução do endereço virtual gerado no exercício (103-a) em um endereço físico.

(105) Quantas páginas terá uma tabela de paginação considerando que o espaço de endereçamento seja de 32 bits e o tamanho da página seja de 4k bytes?

(106) Refaça o exercício (105) considerando páginas de 2k, 8k e 16k bytes.

(107) Devido a memória virtual um acesso a memória acaba se tornando dois acessos a memória pois para a tradução do endereço virtual em endereço físico a MMU precisa consultar a tabela de paginação que deve residir em algum lugar e é muito grande para residir internamente na MMU. Há alguma forma de aliviar esta situação? Descreva o que é a TLB e como ela funciona.