MC514—Sistemas Operacionais: Teoria e Prática

Lista de exercícios III*

Não precisa ser entregue!!

Questões básicas

1. Enumere e explique resumidamente as principais responsabilidades de um sistema operacional.
2. Qual é a diferença entre modo usuário e modo kernel? Por que esta distinção é feita?
3. Descreva como são implementados os seguintes eventos: (i) chamada de sistema, (ii) tratamento de interrupção e (iii) tratamento de sinais. Quais são as semelhanças e diferenças entre estes eventos?
4. Qual é a diferença entre um sistema monolítico e um baseado em camadas?

Processos e Threads

1. Qual é a diferença entre um processo CPU-bound e um processo I/O-bound? É possível determinar qual é o tipo de um processo por meio de uma análise do seu código? E em tempo de execução? Processos I/O bound merecem prioridade mais alta nos algoritmos de escalonamento?
2. *Throughput* é o número de *jobs* por hora que um sistema completa. *Turnaround time* é o tempo médio entre o momento que um *job* é submetido e o momento em que ele termina. Comente como estes fatores são afetados se você adotar uma política de escalonamento do tipo *shortest job first*.
3. Descreva a política de escalonamento *round robin*. Quais análises custo-benefício precisam ser feitas para se estabelecer o intervalo de tempo que um processo deve executar?
4. Suponha que o sistema operacional que você está utilizando só executa um processo quando não existem processos de maior prioridade prontos para rodar. Como esta política pode inviabilizar o uso de espera ocupada por uma determinada condição?
5. Descreva o funcionamento da função `fork()`. Após o `fork`, como os processos pai e filho podem se comunicar/sincronizar?

*Algumas questões desta lista são gerais e enquanto outras são específicas do sistema operacional UNIX.

Gerência de memória

1. Quais são as diferenças básicas entre a utilização de lista de livres e bitmaps para a gerência de memória?
2. Comente as vantagens e desvantagens das políticas *first fit*, *next fit*, *best fit* e *worst fit* para a ocupação de blocos de memória.
3. Qual é a diferença entre endereço virtual e físico? Por que esta distinção é interessante?
4. O que é *swapping*? Quando este mecanismo é usado, é interessante que os programas sejam codificados considerando endereços virtuais ou físicos?
5. Como funcionavam os overlays? Por que você acha que esta estratégia foi abandonada?
6. Explique o funcionamento do mecanismo de paginação. Como é feito o mapeamento entre endereços virtuais e físicos? Por que trabalhar com potências de 2 é interessante?
7. Uma tabela de páginas armazenada inteiramente em memória pode comprometer o desempenho do sistema. Por outro lado, armazenar toda a tabela em registradores pode ser muito caro. Explique como um dispositivo do tipo TLB (Translation Lookaside Buffer) pode ser um bom compromisso entre estas duas situações críticas.
8. Explique como funciona a paginação em vários níveis e quando ela é interessante.
9. Explique como o mecanismo de paginação pode facilitar o compartilhamento de memória entre dois processos distintos.
10. Descreva e comente políticas para substituição de páginas.

Sistema de Arquivos

1. Diretórios armazenam informações sobre arquivos. É interessante armazenar todos os atributos de gerência de um arquivo no diretório?
2. O que é um i-node?
3. O que são *hard links* e *symbolic links*? O uso de i-nodes facilita a implementação de hard links? E de symbolic links?
4. Quando um sistema de arquivos baseado em i-nodes pode ficar em estado inconsistente? Descreva o objetivo e o funcionamento do programa FSCK (file system check).
5. Explique como funciona o mecanismo de *caching* de blocos do disco baseado em *hashing* e política LRU (least recently used).
6. O que são *pipes*? Por que pode ser mais interessante utilizar um pipe do que um arquivo intermediário?

Entrada e Saída

1. O que são *device drivers*? Como os device drivers podem ser acoplados a um sistema operacional?

2. A CPU se comunica com os dispositivos de entrada e saída por meio de registradores especiais presentes nos controladores. Explique quais são as vantagens de se usar um espaço de endereçamento separado para estes registradores ou usar o mesmo espaço de endereçamento da memória (memory-mapped I/O).

3. Muitas vezes, palavras da memória podem ficar temporariamente armazenadas em um cache para aumentar o desempenho dos programas. Faria sentido fazer também o caching dos registradores dos dispositivos?

4. Comente as diferenças entre um código para acesso aos dispositivos de entrada e saída baseado em espera ocupada e outro baseado em interrupções.

5. Explique o funcionamento do esquema denominado DMA (Direct Memory Access).

Seções selecionadas do livro :-)

Abaixo estão enumeradas algumas seções do livro Modern Operating Systems, Andrew S. Tanenbaum, Prentice Hall, 2001, segunda edição, que foram comentadas em sala de aula.

- Capítulo 1 (Introdução): 1.1, 1.4, 1.5 e 1.6
- Capítulo 2 (Processos): 2.1–2.5
- Capítulo 4 (Gerência de Memória): 4.1–4.4, 4.5.1, 4.6.1, 4.8.1–4.8.2
- Capítulo 5 (Entrada e Saída): 5.1, 5.2, 5.3.1
- Capítulo 6 (Sistema de Arquivos): 6.1–6.2

Veja também a documentação on-line sobre memória compartilhada, mapeamento de arquivos em memória e pipes.