Exercícios sobre Processos e Threads

Lista04

Informações

- Entrega pelo Portal COMP em <http://trab.dc.unifil.br/moodle/>.

1. Suponha que você fosse projetar uma arquitetura de computador avançada que realizasse chaveamento de processos em hardware, em vez de interrupções. De qual informação a CPU precisaria? Descreva como o processo de chaveamento por hardware poderia funcionar.

2. Em todos os computadores atuais, pelo menos parte dos tratadores de interrupções é escrita em linguagem de montagem. Por quê?

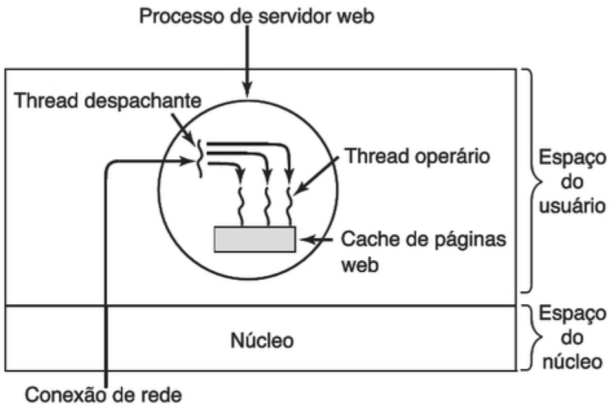
3. Quando uma interrupção ou uma chamada de sistema transfere o controle para o sistema operacional, geralmente uma área da pilha do núcleo separada da pilha do processo interrompido é usada. Por quê?

4. Múltiplas tarefas podem ser executadas em paralelo e terminar mais rápido do que se forem executadas de modo sequencial. Suponha que duas tarefas, cada uma precisando de 20 minutos de tempo da CPU, iniciassem simultaneamente. Quanto tempo a última levará para completar se forem executadas sequencialmente? Quanto tempo se forem executadas em paralelo? Presuma uma espera de E/S de 50

5. Considere um sistema multiprogramado com seis programas na memória ao mesmo tempo. Presuma que cada processo passe 40% do seu tempo esperando pelo dispositivo de E/S. Qual será a utilização da CPU?

6. Presuma que você esteja tentando baixar um arquivo grande de 2 GB da internet. O arquivo está disponível a partir de um conjunto de servidores espelho, cada um deles capaz de fornecer um subconjunto dos bytes do arquivo; presuma que uma determinada solicitação especifique os bytes de início e fim do arquivo. Explique como você poderia usar os threads para melhorar o tempo de download.

7. Um servidor web multithread é mostrado na figura a seguir. Se a única maneira de ler de um arquivo é a chamada de sistema `read` com bloqueio normal, você acredita que threads de usuário ou threads de núcleo estão sendo usados para o servidor web? Por quê?



8. Na tabela a seguir, o conjunto de registradores é listado como um item por thread em vez de por processo. Por quê?

Itens por processo	Itens por thread
Espaço de endereçamento	Contador de programa
Variáveis globais	Registradores
Arquivos abertos	Pilha
Processos filhos	Estado
Alarmes pendentes	
Sinais e tratadores de sinais	
Informação de contabilidade	

9. Por que um thread em algum momento abriria mão voluntariamente da CPU chamando emphthread_yield?

10. Qual é a maior vantagem de se implementar threads no espaço de usuário? Qual é a maior desvantagem?

11. No código a seguir as criações dos threads e mensagens impressas pelos threads são intercaladas ao acaso. Existe alguma maneira de se forçar que a ordem seja estritamente thread 1 criado, thread1 imprime mensagem, thread 1 sai, thread 2 criado, thread 2 imprime mensagem, thread 2 sai e assim por diante? Explique sua resposta.

```
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  #define NUMBER_OF_THREADS 10
6
7  void *print_hello_world(void *tid) {
8      /* Esta funcao imprime o identificador do thread e sai. */
```

```
9       printf("Ola mundo. Boas vindas do thread %d\n", tid);
10      pthread_exit(NULL);
11  }
12
13  int main(int argc, char *argv[]) {
14      /* O programa principal cria 10 threads e sai. */
15      pthread_t threads[NUMBER_OF_THREADS];
16      int status, i;
17
18      for (i = 0; i < NUMBER_OF_THREADS; i++) {
19          printf("Metodo Main. Criando thread %d\n", i);
20          status = pthread_create(
21              &threads[i], NULL, print_hello_world, (void*)i);
22
23          if (status != 0) {
24              printf("Oops. pthread_create retornou erro %d\n", status);
25              exit(-1);
26          }
27      }
28      exit(NULL);
29  }
```