1) Quais são as diferenças essenciais entre um aplicativo comum (por

exemplo, o Microsoft Word) e um sistema operacional?

A diferença é que é o programa mais importante é a parte que fica entre o hardware e os programas aplicativos, então o sistema operacional que carrega os programas e aplicativos.

2) Por que um sistema operacional é necessário?

Para permitir o compartilhamento dos recursos de hardware por diversas aplicações além de gerenciar os recursos do sistema.

3) Que nome é dado às chamadas realizadas pelas aplicações aos serviços

disponibilizados pelo sistema operacional? Mencione como exemplo duas

destas chamadas.

São chamadas de Chamadas de Sistema (System Calls) , dois exemplos de chamadas de sistemas são “access()” e “listen()”.

4) O que é um processo?

É um **"trabalho (programa)"** em execução em um sistema operacional.

5) Um processador com um único núcleo executa vários processos

simultaneamente? Explique sua resposta.

Não, pois para existir processos executados simultaneamente, deve existir dois ou mais núcleos de processamento. Portanto, com um único núcleo, deve ocorrer o compartilhamento de tempo de uso do processador entre vários processos. Em processadores de um só núcleo, as funções de multitarefa podem ultrapassar a capacidade da CPU, o que resulta em queda no desempenho enquanto as operações aguardam para serem processadas.

6) O que ocorre quando o sistema operacional faz o chaveamento entre

processos?

O sistema operacional salva todo o estado do processo que está em execução, em uma estrutura chamada PCB (Process Control Block), e carrega o estado do processo que entrará em execução.

7) Qual é a diferença entre thread e processo?

Thread: é a execução de instruções criadas pelo programa,e é executada em um único local, diferente de um processo que possui um espaço reservado na memória.

Processo: é a execução do programa, sequências de instruções;

8) Qual é a diferença entre escalonamento preemptivo e não-preemptivo.

Preemptivo:

* não depende somente do processo;
* S.O. pode tomar o processador mesmo que ainda tenha processos a executar;

Não-Preemptivo:

* depende somente do processo;
* processador não pode ser tomado;

9) Cite 2 algoritmos de escalonamento de processos e explique o funcionamento de cada um.

First-Come,First-Served (FCFS) : os processos são executados na ordem em que chegam ao escalonador. Pode apresentar diferentes tempos médios de espera de acordo com a ordem de chegada dos processos.

-Shortest JobFirst (SJF):

* Versão não-preemptiva: uma vez que o processo recebe tempo de CPU, não pode ser interrompido até que utilize o tempo de CPU alocado.
* Versão preemptiva: se um novo processo chega e possui um tempo de CPU menor que o tempo restante do processo em execucão, este será interrompido. Esta forma é conhecida como Shortest-Remaining-TimeFirst (SRTF).

10) Cite 2 recursos disponibilizados pelo sistema operacional para ajudar o

programador a implementar programas concorrentes sem interferência entre

threads, e diga em qual situação é mais conveniente utilizar cada recurso.

Cooperação: threads trabalham em cooperação para atingir um objetivo.

Ex:

Semáforo Mecanismos que garantem que somente N processos podem acessar um   certo recurso em um dado momento.

       Competição: threads concorrentes disputam pelo uso exclusivo de recursos.

Ex:

Combinação de transferência de dados com exclusão mútua e sincronização. Um processo gera as mensagens (produtor), colocando-as em uma fila de mensagens. O outro processo (consumidor) retira as mensagens da fila. Caso a fila fique totalmente cheia, o processo produtor espera até que seja liberado espaço na fila. Caso a fila fique vazia, o processo consumidor espera até que um novo elemento seja inserido na fila.