* 1
  + Gerenciador de Recursos:
    - Pode ser chamado assim pois prove aos usuários chamadas ao sistema, abstraindo a necessidade de acessar diretamente o Hardware, o Kernel toma conta dessa parte.  
      Ao mesmo tempo, controlando processador, memorias, hard-drives e outros recursos. Por isso é chamado de “Gerenciador de Recursos”.
  + Máquina virtual ou estendida:
    - Desenvolver ou enviar comandos diretamente para o Hardware é uma tarefa complexa. Por exemplo, para lermos um disquete, é necessário saber os comandos para liga/desliga a leitora, leia cada setor do disquete, entre outros, e para cada comando deste, existem muitas vezes dezenas de parâmetros.  
      O SO é considerado uma maquina virtual pois, para facilitar nossa vida, abstrai essa complexidade, por exemplo, em suas próprias system calls. Não queremos saber qual setor, formato, maneira que queremos gravar o dado no disquete, o próprio write() toma conta disso.
* 2
  + Podem ser chamadas dessa maneira pois nos sistemas que suportam multitrheading elas consegue compartilhar dados na memória. Dessa maneira, seu processamento é mais simples e rápido.
* 3
  + Uma thread alterar uma variável local de outro processo ou thread pode causar grande instabilidade no processo/thread a ser executado. Pode até mesmo que essa thread tente acessar essa variável local enquanto outro processo já esta tentando acessá-la, também causando instabilidade. Por isso threads não compartilham variáveis locais, somente globais. Nas globais existem sistemas que podem evitar esses tipos de erro.  
    Sem contar que variáveis locais nem sequer são acessíveis o tempo todo por outras threads, pois não estão o tempo todo em execução, como as variáveis globais.
* 4
  + Os próprios nomes já são intuitivos
    - Gerenciamento de Processos:
      * O SO gerencia a inicialização, o encerramento, o funcionamento, atende as necessidades dos processos como por exemplo as sys calls e também pode integrar a comunicação entre um processo e outro
    - Gerenciamento de Entrada/Saida
      * O SO torna-se responsável por gerenciar os dispositivos conectados à maquina. Seja teclado, mouse, monitor ou impressora. São os “devices”, periféricos.
    - Gerenciamento de Memória:
      * O SO gerencia a memória, alocando-a de acordo com a necessidade de cada ação ou software executado no sistema. Nenhum processo tem acesso direto à memória, isso poderia causar grandes instabilidades no sistema. Os softwares fazem requisições ao SO e só então o SO se responsabiliza em manejar a memória
    - Sistemas de Arquivos
      * É através deste que ocorre a gravação e leitura dos dados em qualquer dispositivo de armazenamento em um computador. Atualmente existem mais de 20 sistemas de arquivos suportados pelo Linux
* 5
  + Deve armazenar essa tabela para saber como executar esse processo; saber, durante o escalonamento, aonde o processamento foi pausado etc.
* 6
  + Executando: Atualmente em execução pelo processador
  + Parado: Aguardando para ser executado
  + Bloqueado: Bloqueado para a execução por algum motivo
* 7
  + Condições de disputa ou Race Conditions é uma ocasião onde duas threads acessam uma variável global ao mesmo tempo. Quando isto ocorre, as duas threads executam seus respectivos comandos, gerando dados inconsistentes, pois, a variável global mutualmente acessada foi alterada por outra thread durante a execução.
* 8
  + A vantagem de se utilizar o sleep/wakeup ao invés da espera ocupada é que, no primeiro caso, não há possibilidade de ocorrer a inversão de prioridade, onde um processo de maior prioridade trava um de menor prioridade em sua região critica
* 9
  + É chamada hibrida pois é monolítica e microkernel ao mesmo tempo. Parte do sistema funciona de uma maneira, outra parte de outra.  
    Existem drivers em espaço de usuário e drivers em espaço de kernel.
* 10
  + O SO prove o gerenciamento de todo o hardware presente na maquina através do gerenciamento de recursos, como por exemplo as sys calls. Esses recursos são:
    - Processos
    - Memoria
    - Arquivos
    - Entrada/Saida
  + Entre outros.
* 11
  + Monolítica
    - Todo os drivers controladores do hardware estão presentes dentro do espaço de kernel.
      * Vantagem:
        + Desempenho, pois como todos os drivers já estão em espaço de kernel, sua execução é facilitada
      * Desvantagem:
        + Se algum erro ocorrer, há o risco de todo o sistema travar, pois sua execução está dentro do espaço de kernel.
  + Micro-Kernel
    - Todos os drivers controladores do hardware estão em espaço de usuário
      * Vantagem:
        + Se algum erro ocorrer, o processo do driver pode ser simplesmente encerrado e reinicializado como qualquer outro processo de usuário. Sem comprometer a estabilidade do sistema
      * Desvantagem:
        + Como sua execução se deve dentro do espaço de usuário, todo o processo é mais lento quando comparado aos sistemas utilizadores do sistema monolítico.
* 12
  + Espaço de Kernel
    - É o espaço na memória reservado ao kernel e os drivers de dispositivo (se o sistema for monolítico) . O usuário nunca acessa diretamente esse espaço na memória. O acesso é feito através das sys calls
  + Espaço de Usuário
    - É o espaço na memória reservado ao usuário e seus processos. Esse espaço também pode ser utilizado para os drivers de dispositivos se o sistema for micro-kernel.
* 13
  + São chamadas efetuadas pelos processos de usuário ou até mesmo por drivers em espaço de kernel. Essas chamadas têm por função evitar que processos acessem diretamente o hardware.   
    As chamadas do sistema fazem parte do kernel. Se qualquer informação necessária para um processo realizar suas atividades necessite de uma informação provida diretamente do hardware, essa informação será provida pelo Kernel através de uma Chamada de Sistema ou System Calls.
* 14
  + Como Tanenbaum explica (e eu não faço a mínima ideia de como implementar isso em código) é possível solucionar o problema com um matriz de semáforos, um por filosofo, assim os filósofos com fome podem bloquear se os garfos necessários estiverem ocupados.
* 15
  + O escalonador do SO é o dispositivo de gerenciamento de processos que permite que o SO execute mais de um processo por vez. Ele escalona entre um processo e outro, sempre pausando o estado do processo atual para iniciar ou continuar o próximo, assim, podendo periodicamente (mas não acintosamente) voltar a executar o processo pausado anteriormente sem que haja perda de dados ou status de processamento
* 16
  + O escalonador deve seguir algum critério para escalonar os processos, como por exemplo:
    - Ser justo: Todos os processos tem acesso e processamento iguais
    - Maximizar a produtividade: o escalonador escolhe os processos dos quais aparentemente vão agilizar a produtividade do sistema
    - Entre vários outros... Existem vários tipos de critérios
    - Roun Robin citado em sala de aula: reserva uma quantidade de tempo de execução para cada processo e pula para o próximo, passando por todos os processos de maneira circular
* 17
  + ???
  + Não faço a mínima ideia
* 18
  + São métodos de comunicação entre processos
    - FIFO: pipes onde o output de um comando entra como input de outro
    - Memória compartilhada: os processos se comunicam através de partes de memoria compartilhada
    - Sockets: parecidos com os FIFO (pipes) mas mais adequados para comunicação cliente/servidor
* 19
  + Pelo motivo óbvio que, se está interrompido o acesso, é porque está em uso. Acessar simultaneamente parte da memória de região crítica é extremamente arriscado pois pode comprometer, algumas vezes, toda a estabilidade do sistema.
* 20
  + São softwares específicos de comunicação e configuração dos Hardwares como:
    - Driver de placa de rede
    - Driver de placa de som
    - Driver de placa de vídeo
    - Driver de mouse
    - Driver de teclado
    - Etc.
* 21
  + A vantagem se deve que o processamento é muito mais rápido quando comparado ao espaço de usuário
* 22
  + São barramentos físicos de hardware que permitem a comunicação e conexão físicas de dispositivos externos ao sistema:
    - VESA
    - PCI
    - PCI-E
    - USB
    - IRDA
    - PCMCIA
    - Etc