CONCURSO PÚBLICO
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO DE JANEIRO

EDITAL Nº 006/2022

PADRÃO DE RESPOSTAS DA PROVA DISCURSIVA REALIZADA DOMINGO, 15 DE MAIO DE 2022.
PRAZO PARA RECURSO CONTRA O PADRÃO DE RESPOSTAS: 16 E 17 DE MAIO DE 2022, NO ENDEREÇO ELETRÔNICO:
http://www.selecon.org.br

PADRÃO DE RESPOSTAS PRELIMINAR

SJM – 02

INFORMÁTICA
Sistemas Operacionais; Infraestrutura de Rede de Computadores; Segurança da Informação

Nº DA QUESTÃO	Espera-se que o candidato(a) desenvolva os aspectos/conteúdos propostos a seguir.
1	O candidato deverá desenvolver o(s) conteúdo(s) com base nos seguintes aspectos: A) Conceitos, tipos e vantagens do serviço de DHCP **(2,5 pontos)** B) Os processos de funcionamento do DHCP (Etapas de concessão) **(5,0 pontos)** C) A Renovação do endereço IP **(2,5 pontos)** Total previsto de linhas para a resposta final do(a) candidato(a): **25 a 30 linhas**

Protocolo DHCP

O protocolo DHCP é um protocolo de cliente/servidor que fornece automaticamente um *host ip* (protocolo IP) com seu endereço IP e outras informações de configuração relacionadas, como a máscara de sub-rede e o *gateway* padrão. As RFCs 2131 e 2132 definem o DHCP como um padrão de IETF (Força-Tarefa de Engenharia da Internet) com base no protocolo BOOTP, um protocolo com o qual o DHCP compartilha muitos detalhes de implementação. O DHCP permite que os *hosts* obtenham informações de configuração TCP/IP necessárias de um servidor DHCP.

Por que usar o DHCP?

Cada dispositivo em uma rede baseada em TCP/IP deve ter um endereço IP *unicast* exclusivo para acessar a rede e seus recursos. Sem o DHCP, os endereços IP para novos computadores ou computadores que são movidos de uma sub-rede para outra devem ser configurados manualmente; os endereços IP para computadores removidos da rede devem ser recuperados manualmente.

Com o DHCP, todo esse processo é automatizado e gerenciado centralmente. O servidor DHCP mantém um *pool d*e endereços IP e arrenda um endereço para qualquer cliente habilitado para DHCP quando ele é iniciado na rede. Como os endereços IP são dinâmicos (concessões) em vez de estáticos (atribuídos permanentemente), os endereços que não estão mais em uso são retornados automaticamente ao *pool* para realocação.

O administrador de rede estabelece servidores DHCP que mantêm informações de configuração TCP/IP e fornecem a configuração de endereço para clientes habilitados para DHCP na forma de uma oferta de concessão. O servidor DHCP armazena as informações de configuração em um banco de dados que inclui:

- Parâmetros de configuração TCP/IP válidos para todos os clientes na rede.

- Endereços IP válidos, mantidos em um pool para atribuição a clientes, bem como endereços excluídos.

- IP Reservado endereços associados a clientes DHCP específicos. Isso permite a atribuição consistente de um único endereço IP para um único cliente DHCP.

- A duração da concessão ou o período de tempo para o qual o endereço IP pode ser usado antes que uma renovação de concessão seja necessária.

Objetivo

No ambiente TCP/IP, cada *host* deve ter um conjunto de parâmetros configurados para poder comunicar-se. Estes incluem endereço IP, máscara*, default gateway,* servidor de DNS e outros. Alguns desses parâmetros são mandatórios e outros são opcionais.

Essencialmente, quando uma máquina inicializa, o DHCP cliente envia pela rede uma solicitação de endereço IP (e outros parâmetros) e informando o endereço físico da máquina (MAC Address). Um DHCP *server* responde a esta solicitação, enviando a configuração solicitada.

Terminologia

- **DHCP server**: *host* que provê os endereços aos clientes através da porta UDP 67

- **DHCP client**: *host* que solicita endereços, usado a porta UDP 68.

- **DHCP relay**: roteador ou *host q*ue propaga as solicitações de um *client* a um *server* que está em outra rede e vice-versa;

Modos de Operação

O DHCP pode operar de três maneiras:

- **Manual**: neste modo, o administrador deverá listar todos os MAC *address* dos clientes que serão atendidos pelo DHCP *server*, e designar a cada um deles um endereço IP. Solicitações de outros *clients* não serão atendidas. É uma configuração importante para *hosts* que devem ter endereços fixos, como roteadores, servidores, etc; e também como procedimento de segurança, já que, deste modo, uma máquina não prevista não poderá conectar-se à rede por este meio.

- **Automática**: o administrador configura uma faixa de endereços IP, que serão fornecidos à medida em que forem solicitados pelos clientes. Uma vez fornecidos, estes endereços ficam permanentemente designados ao cliente, a menos que o administrador realize uma intervenção manual.

- **Dinâmica**: nesta configuração, que é a mais usada, uma faixa de endereços é configurada e, dela, um endereço será emprestado ao solicitante por um período, chamado lease period. À medida que não forem mais necessários, esses endereços tornam-se novamente disponíveis a outros clientes automaticamente, facilitando o trabalho administrativo. O *pool* de endereços deverá corresponder apenas à quantidade de máquinas que operarem simultaneamente e não a todas as máquinas existentes. É uma solução importante para provedores de acesso, por exemplo.

No início da operação do DHCP, quatro pacotes são trocados, conforme ilustrado a seguir:

DHCPDISCOVER

O cliente envia um pacote *broadcast* DHCPDISCOVER na porta 68 indicando que necessita de endereço IP e outros parâmetros. No pacote, o endereço de destino é 255.255.255.255 e o endereço de origem é 0.0.0.0. A seção DHCP identifica o pacote como um pacote Discover e identifica o cliente em dois locais usando o endereço físico da placa de rede (MAC Address). O cliente está no status **Init**.

DHCPOFFER

O servidor DHCP responde enviando um pacote DHCPOFFER *unicast* remetido ao MAC Address do cliente. Na seção IP, o endereço de origem agora é o endereço IP do servidor DHCP e o endereço de destino é o endereço de difusão 255.255.255.255.255. O campo DHCP identifica o pacote como uma Oferta. O campo LTDADDR é preenchido com o **endereço IP** que o servidor está oferecendo ao cliente. O campo LTDDR ainda contém o endereço físico do cliente solicitante. O Campo de Opção DHCP as várias opções que estão sendo enviadas pelo servidor junto com o endereço IP. Nesse caso, o servidor está enviando a Máscara de Sub-rede, o Gateway Padrão (Roteador), o Tempo de Concessão, o endereço do servidor WINS (Serviço de Nome NetBIOS) e o Tipo de Nó NetBIOS. Quando o cliente recebe um ou mais DHCPOFFERs, ele passa ao status **Selecting**.

DHCPREQUEST: O cliente responde ao DHCPOFFER enviando um *broadcast* DHCPREQUEST. O endereço de origem do cliente ainda é 0.0.0. Ao enviar esta solicitação, o cliente passa a estar em Requesting.0 e o Destino do pacote ainda é 255.255.255.255.255. O cliente retém 0.0.0.0 porque o cliente não recebeu a verificação do servidor de que não há problema em começar a usar o endereço oferecido. O Destino ainda é transmitido, porque mais de um servidor DHCP pode ter respondido e estar mantendo uma reserva para uma Oferta feita ao cliente. Isso permite que esses outros servidores DHCP saibam que podem liberar seus endereços oferecidos e devolvê-los aos *pools* disponíveis. O campo DHCP identifica o pacote como uma Solicitação e verifica o endereço oferecido usando o campo DHCP: Endereço Solicitado. O campo DHCP: Identificador de Servidor mostra o endereço IP do servidor DHCP que oferece a concessão.

DHCPACK: O servidor DHCP responde ao DHCPREQUEST com um DHCPACK com a informação completa para o cliente. O endereço de origem é o endereço IP do servidor DHCP e o Endereço de destino ainda é 255.255.255.255. O campo LTDADDR contém o endereço do cliente e os campos DEDDD e DHCP: Identificador do Cliente são o endereço físico da placa de rede no cliente solicitante (MAC Address). O campo DHCP identifica o pacote como um **ACK.** A partir deste, inicia-se a contagem do período de empréstimo, e o cliente está em **Boun**.

Renovação do endereço IP

O período de empréstimo (*lease period*) é informado com os demais parâmetros solicitados. Além do valor do período, dois outros valores são enviados, cujos valores *default* são:

T1 = 0,5 x lease-period

T2 = 0,875 x lease-period

Quando T1 expira, o cliente envia um DHCPREQUEST unicast para o servidor, solicitando prorrogação do prazo. Neste ponto, seu status é **Renewing**.

Caso seja autorizado (depende da configuração do server), o servidor enviará um DHCP Acknowledgement, reiniciando a contagem do período de empréstimo, e retornando a **Bound**.

	Caso o cliente não receba um DHCPACK até T2, ele enviará um DHCPREQUEST *broadcast*, que poderá ser respondido por qualquer servidor DHCP, e alterará seu status para **Rebinding**. Caso ele não receba nenhuma nova configuração até o final do *lease period*, ou receba um **DHCPNACK** informado que a solicitação foi rejeitada, ele cessará as comunicações e retornará ao estado inicial, **Init**.
2	O candidato deverá desenvolver o(s) conteúdo(s) com base nos seguintes aspectos: A) Explicar o problema da condição de corrida, destacando o motivo da sua ocorrência e suas possíveis consequências. (3,0 pontos) B) Explicar o que é um semáforo binário e como ele pode ser utilizado para resolver o problema da condição de corrida, destacando o papel das regiões críticas. (3,0 pontos) C) Citar e explicar as condições necessárias para a ocorrência de deadlocks (2,0 pontos) D) Identificar a ocorrência de deadlocks utilizando grafos de alocação de recursos (2,0 pontos) Total previsto de linhas para a resposta final do(a) candidato(a): **entre 30 e 40 linhas** a) Suponha que os recursos compartilhados R1, R2 e R3 sejam variáveis globais no espaço de endereçamento virtual do processo P. Nesse caso, uma vez que as threads de um processo compartilham seu espaço de endereçamento virtual, T1, T2 e T3 podem ler e escrever livremente em R1, R2 e R3. Suponha também que em dado momento a thread T1 queira incrementar a variável R1 e que a thread T2 queira decrementar a variável R1. As funções de incrementar e decrementar R1 executam as seguintes instruções no processador: **Incrementar R1** ● Mover o valor de R1 da memória para um registrador, digamos X1 ● Incrementar o valor de X1 ● Mover o valor de X1 para o endereço de memória de R1 **Decrementar R1**

- Mover o valor de R1 da memória para um registrador, digamos X2
- Decrementar o valor de X2
- Mover o valor de X2 para o endereço de memória de R1

Se as threads T1 e T2 executassem de forma sequencial, ao final de uma execução de cada thread o valor em R1 se manteria o mesmo. Por exemplo, se o valor inicial de R1 fosse 5, a execução de T1 alteraria esse valor para 6, mas a execução de T2 faria com que R1 voltasse ao valor inicial 5. Porém, quando as threads T1 e T2 executam de forma concorrente, a seguinte sequência de execução é possível:

1 - T1: Mover o valor de R1 da memória para um registrador, digamos X1
2 - T2: Mover o valor de R1 da memória para um registrador, digamos X2
3 - T2: Decrementar o valor de X2
4 - T1: Incrementar o valor de X1
5 - T1: Mover o valor de X1 para o endereço de memória de R1
6 - T2: Mover o valor de X2 para o endereço de memória de R1

Se o valor inicial de R1 for 5, após a execução da sequência acima o valor de R1 seria atualizado para 4. Se, por outro lado, alterássemos a ordem das instruções 5 e 6, o valor de R1 seria atualizado para 6. Ou seja, dependendo da ordem em que as threads T1 e T2 são escalonadas pelo sistema operacional, o valor em R1 pode ser 4, 5 ou 6.

Como no exemplo anterior, o problema da condição de corrida ocorre quando recursos compartilhados são acessados concorrentemente por threads e o resultado do seu processamento depende da ordem na qual as instruções dessas threads são executadas. Como a ordem de execução das threads não é determinística, a condição de corrida gera erros de processamento também não determinísticos, tornando-os difíceis de detectar e reproduzir.

b) Regiões críticas são trechos de código em uma thread que acessam recursos compartilhados. Ao permitir que threads acessem suas regiões críticas livremente, criamos as condições necessárias para ocorrência de condições de corrida, conforme descrito no item anterior. Assim, para impedir a ocorrência de condições de corrida, faz-se necessário impedir que mais de uma thread acesse ao mesmo tempo uma região crítica associada ao mesmo recurso compartilhado. Em outras palavras, precisamos garantir exclusão mútua nas regiões críticas.

Um semáforo pode ser entendido como uma variável inteira especial que só pode ser manipulada por três operações:

- Inicialização
- operação up
- operação down

Um semáforo binário é um tipo especial de semáforo utilizado para garantir exclusão mútua na região crítica. Ele deve ser inicializado com o valor 0 ou 1. A operação down é executada antes de uma thread entrar na região crítica. Se o valor do semáforo for zero, a thread que executou a operação down é bloqueada. Se, por outro lado, o valor do semáforo for 1, seu valor é decrementado e a thread que executou a operação down pode continuar e entrar na região crítica. Quando uma thread executa a operação up o valor do semáforo é incrementado. Nesse caso, se o valor do semáforo continuar menor ou igual a zero, uma thread bloqueada no semáforo é acordada e sua entrada na região crítica é permitida.

Assim, se queremos garantir exclusão mútua na região crítica R, podemos utilizar um semáforo binário s da seguinte maneira:

INICIALIZAÇÃO: s = 1

down(s)
R
up(s)

Por fim, note que as operações up e down devem ser executadas de maneira atômica. Caso contrário, a utilização do semáforo também pode causar uma condição de corrida.

c) Ao resolver o problema da condição de corrida, é possível levar o sistema a uma condição de deadlock. Cite e explique as condições necessárias para a ocorrência de deadlocks.

As condições necessárias são:

1. Exclusão Mútua: O sistema possui recursos que só podem ser acessados por uma thread por vez (como semáforos, por exemplo);
2. Não preempção: Uma vez que uma thread/processo tem um recurso alocado pelo sistema, esse recurso só é liberado voluntariamente pela thread/processo ou quando a thread/processo termina;
3. Obtém e Espera: uma thread pode ter um recurso alocado enquanto aguarda pela alocação de outros recursos que, por sua vez, estão alocados para outras threads
4. Espera circular: Deve existir um conjunto de threads em espera $\{T_0, T_1, \ldots, T_n\}$ tal que T_0 aguarda um recurso alocado para T_1, que aguarda um recurso alocado para T_2, ..., T_{n-1} aguarda um recurso alocado para T_n que, por sua vez, aguarda um recurso alocado para T_0.

É importante notar que todas as condições devem ser satisfeitas para que um deadlock possa ocorrer.

d) Uma vez que os recursos descritos no grafo de alocação são semáforos, que por sua vez representam regiões críticas, podemos dizer que as três primeiras condições necessárias para ocorrência de um deadlock são satisfeitas:

	1. Exclusão Mútua: podemos dizer que quando a região crítica é acessada por uma thread, o semáforo é alocado pela thread e quando uma thread executa a operação down em um semáforo com valor zero, ela requisita a alocação desse semáforo, mas não a obtém imediatamente. Assim, como o acesso à região crítica é feito de forma a garantir exclusão mútua, o mesmo podemos dizer sobre a alocação dos semáforos. 2. Não preempção: uma vez que uma thread obtém acesso a uma região crítica, ela só perderá esse acesso ao deixar a região crítica voluntariamente. 3. Obtém e Espera: não há nada que impeça uma thread de executar a operação down em dois semáforos diferentes. Se no primeiro semáforo a thread obtiver acesso a região crítica, mas no segundo semáforo a thread bloquear, teremos a situação de obtém e espera. Assim, para definirmos se há ou não deadlock, precisamos verificar se a quarta condição, espera circular, é ou não satisfeita nesse caso. Como semáforos tem apenas uma instância, podemos identificar uma espera circular em um grafo de alocação se este apresentar um ou mais ciclos. No nosso caso, o seguinte ciclo pode ser identificado: T3 -> S3 -> T2 -> S3 -> T3. Portanto, podemos afirmar que o sistema está em um estado de deadlock.
3	O candidato deverá desenvolver o(s) conteúdo(s) com base nos seguintes aspectos: A) Os 3 principais conceitos de segurança de computadores: confiabilidade, integridade e disponibilidade e exemplos práticos **(5,0 pontos)** B) Descrever o tipo de ataque **(2,5 pontos)** C) Identificar o princípio de segurança de computadores atingido com o ataque descrito **(2,5 pontos)** **Segurança de computadores** é a proteção oferecida para um sistema de informação automatizado a fim de alcançar os objetivos de preservar a integridade, a disponibilidade e a confidencialidade dos recursos do sistema de informação (incluindo *hardware*, *software*, *firmware*, informações/dados e telecomunicações). Essa definição introduz três objetivos principais que são o coração da segurança de computadores: **Confidencialidade**: preservar restrições autorizadas sobre acesso e divulgação de informação, incluindo meios para proteger a privacidade de indivíduos e informações privadas. Uma perda de confidencialidade seria a divulgação não autorizada de informação. Esse termo cobre dois conceitos relacionados: **Confidencialidade de dados**: assegura que informações privadas e confidenciais não estejam disponíveis nem sejam reveladas para indivíduos não autorizados.

	Privacidade: assegura que os indivíduos controlem ou influenciem quais informações relacionadas a eles podem ser obtidas e armazenadas, da mesma forma que como, por quem e para quem essas informações são passíveis de ser reveladas. Exemplo de confidencialidade: informações relacionadas às notas devem ser somente disponibilizadas para os estudantes, seus pais e funcionários que precisem delas para realizar o seu trabalho. Dados de matrícula dos estudantes podem se amparar em uma confidencialidade moderada. **Integridade**: prevenir-se contra a modificação ou destruição imprópria de informação, incluindo a irretratabilidade e autenticidade dela. Uma perda de integridade seria a modificação ou destruição não autorizada de informação. Esse termo abrange dois conceitos relacionados: **Integridade de dados**: assegura que as informações e os programas sejam modificados somente de uma maneira especificada e autorizada. **Integridade do sistema**: assegura que um sistema execute as suas funcionalidades de forma ilesa, livre de manipulações deliberadas ou inadvertidas do sistema. exemplo de integridade: muitos aspectos da integridade são ilustrados pelo exemplo das informações das alergias dos pacientes de um hospital armazenadas em um banco de dados. O médico precisa confiar que elas estão corretas e atualizadas. **Disponibilidade:** assegurar acesso e uso rápido e confiável da informação. Uma perda de disponibilidade é a perda de acesso ou de uso da informação ou sistema de informação. Assegurar que os sistemas operem prontamente e seus serviços não fiquem indisponíveis para usuários autorizados. Exemplo de disponibilidade: quanto mais crítico for um componente ou serviço, maior será o nível de disponibilidade requerido. Considere um sistema que oferece serviços de autenticação para sistemas, aplicações e dispositivos críticos. Uma interrupção do serviço resulta na incapacidade dos clientes de acessar os recursos computacionais e de pessoal a fim de chegar ao que necessitam para realizar tarefas críticas. **Ataque à Segurança:** pode ser classificado como os termos de ataques passivos e ataques ativos. Um ataque passivo tenta descobrir ou utilizar informações do sistema, mas não afeta os seus recursos. Um ataque ativo tenta alterar recursos do sistema ou afetar sua operação. O ataque descrito no cenário é um ataque ativo que envolve alguma modificação do fluxo de dados ou a criação de um fluxo falso, que é categorizado como negação de serviço. **A negação de serviço:** impede ou inibe o uso ou gerenciamento normal das instalações de comunicação. Esse ataque pode ter um alvo específico: por exemplo, uma entidade a suprimir todas as mensagens dirigidas para determinado destino (por exemplo, um

	servidor *web*). Outra forma de negação de serviço é a perturbação de uma rede inteira, seja desativando-a ou sobrecarregando-a com mensagens, a fim de prejudicar seu desempenho. O conceito de segurança de computadores que ficou vulnerável com o ataque foi a **disponibilidade**. Total previsto de linhas para a resposta final do(a) candidato(a): **25 a 30 linhas**