

# Concurso Público



## Analista de Sistemas

Ambiente GRID para  
física experimental  
de altas energias

Caderno de Questões  
Prova Objetiva

# 2015

**SRH** SUPERINTENDÊNCIA  
DE RECURSOS  
HUMANOS  
DA UERJ



01|

A rápida absorção das tecnologias de *grid*, por volta de 2001, levou a uma iniciativa baseada, entre outros itens, na identificação de funções essenciais em uma *grid* e no alinhamento dos protocolos do GlobusToolkit (GT) com a tecnologia de serviços Web - o que gerou a linguagem *Web Services Definition Language* (WSDL). Dessa forma, tal iniciativa produziu uma arquitetura que tem em seu conteúdo uma infraestrutura padronizada que possibilitou a definição de um conjunto de interfaces para controlar como os serviços de *grid* devem ser criados, eliminados, nomeados, monitorados etc.

A sigla que denomina o nome dessa infraestrutura é:

- a) OGF
- b) IETF
- c) OGSJ
- d) OGSA

02|

Um usuário pesquisador tem um programa (analisador) que recebe como argumento o caminho de um arquivo e faz sua leitura e processamento. O que é feito no processamento é irrelevante. O usuário tem no diretório /analise milhares de arquivos a serem processados. Os arquivos em questão possuem a extensão ".entrada". Não existem diretórios dentro de /analise. Para executar a análise do arquivo a1.entrada, o usuário deve entrar no diretório /analise e digitar o comando: analisador a1.entrada > /res/a1.entrada.resultados

O resultado da análise é então gravado no arquivo a1.entrada.resultados, na pasta /res. Como são muitos arquivos a serem analisados, o usuário precisa automatizar, com um *script* para *shell*, a tarefa de disparar o programa para todos os arquivos (com extensão .entrada) da pasta de onde o usuário executa o *script*. O pesquisador usa um *shell bash*.

O código do *script* que realiza esse procedimento é:

- a) for arq in \* { analisador \${arq} > /res/\${arq}.resultados }
- b) for arq in \*.entrada ; analisador \$arq > /res/\$arq.resultados
- c) for arq in \* ; do analisador \${arq} > /res/\${arq}.resultados; done
- d) for arq in \*.entrada ; do analisador \$arq > /res/\$arq.resultados; done

03|

Existem três faixas de endereços de IP reservadas para NAT. São exemplos de endereços de IP contidos nessas faixas:

- |                  |              |                |
|------------------|--------------|----------------|
| a) 11.0.0.1      | 172.17.1.1   | 192.168.0.1    |
| b) 10.0.0.1      | 172.16.0.1   | 192.160.0.1    |
| c) 10.255.1.15   | 172.17.255.1 | 192.168.230.25 |
| d) 10.255.230.15 | 172.15.255.1 | 192.168.255.25 |

04|

O controle dos dispositivos de Entrada e Saída (E/S) conectados aos computadores é uma fonte de preocupação dos projetistas de Sistemas Operacionais (SOs). Grande parte dessa preocupação se deve à grande variedade de funções, velocidades e características desses dispositivos. Os SOs atuais definiram um modelo para resolver esse problema.

A opção que explica corretamente como o SO e os dispositivos de E/S interagem é:

- a) os SOs veem somente o subsistema do *kernel* de E/S; este subsistema se encarregará de se comunicar com o *driver* de dispositivo na etapa seguinte
- b) um driver para um dispositivo de E/S pode ser inserido no núcleo do SO, por meio da execução do programa no espaço de usuário e, na sequência, do reinício do SO
- c) para que um dispositivo de E/S seja usado adequadamente, um *driver* correspondente deve ser instalado dentro do SO para que seja executado em modo usuário
- d) em um SO, um *driver* de dispositivo de E/S é o programa que faz a comunicação direta entre a CPU e o componente de *hardware*, por meio de envio e recebimento de comandos

05|

Os parâmetros do comando *iptables* que exibem e apagam todas as regras do *firewall* são, respectivamente:

- a) -S e -D
- b) -L e -F
- c) --list e --delete
- d) --show e --flush

06|

Considere o trecho do programa Python abaixo:

```
def dobra (y):  
    x = y + y  
    return x
```

```
x = 5  
dobra(x)  
dobra(x)  
print x
```

O valor impresso ao executarmos o programa é:

- a) 5
- b) 10
- c) 15
- d) 25

07|

Um sistema tolerante a falhas significa que ele pode prover os serviços, mesmo na presença de falhas. Essas falhas podem ser classificadas em:

- a) síncronas, permanentes e transientes
- b) inerentes, síncronas e independentes
- c) intermitentes, independentes e inerentes
- d) transientes, intermitentes e permanentes

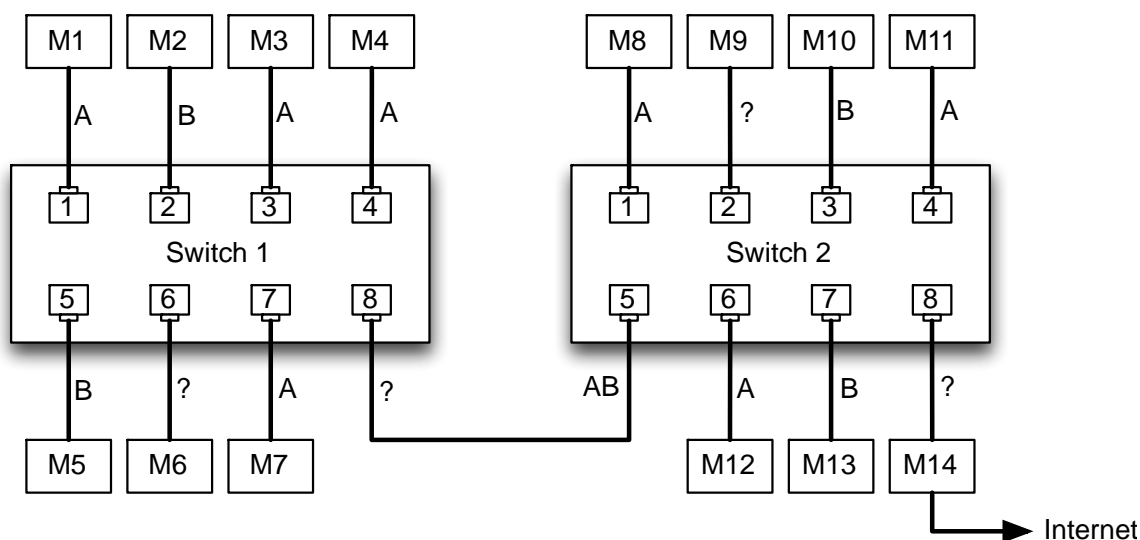
08|

As portas padrão dos protocolos de serviços de rede HTTPS, POP-3, SMTP e SSH são, respectivamente:

- a) 80 / 543 / 631 / 21
- b) 443 / 110 / 25 / 22
- c) 8080 / 22 / 20 / 25
- d) 127 / 443 / 404 / 110

09|

Dois laboratórios compartilham a mesma infraestrutura de rede e decidiram configurar duas VLANs A e B, uma para cada laboratório. As máquinas M1, M3, M4, M6, M7, M8, M11 e M12 devem ficar na VLAN A, e as máquinas M2, M5, M9, M10 e M13, na VLAN B. Foi adquirido um *gateway* (M14) que permitirá a conexão à internet para todas as máquinas da rede, além de dois *switches* de 8 portas que permitem configuração de VLAN. O administrador de rede já fez a ligação de todos os cabos de forma correta, como mostra a figura abaixo:



O administrador também iniciou a configuração das VLANs nas portas dos dois *switches*, faltando configurar apenas as portas 6 e 8 do *switch1* e portas 2 e 8 do *switch2*.

A configuração de VLAN para essas portas são, respectivamente:

- a) A / A / B / B
- b) B / A / A / B
- c) A / AB / B / AB
- d) B / AB / A / AB

10|

O compartilhamento de recursos em um ambiente de *grid* envolve o acesso direto a computadores, dados, *softwares* e outros. Nesse caso, é necessário que se defina o que é compartilhado, quem tem direitos de acesso aos recursos e as condições para que isso ocorra.

Tais regras de compartilhamento são definidas como:

- a) organização virtual
- b) mapeamento de identidade
- c) tradução de credencial de acesso
- d) autenticação em grupo e troca de chave



11|

O protocolo GridFTP é um componente surgido no contexto do GT , sendo uma extensão do conhecido File Transfer Protocol (FTP).

A sua função é:

- a) especificar os componentes lógicos de *grid*
- b) agir como serviço Web aderente aos padrões OGSi
- c) gerenciar operações de dados em ambiente de *grid*
- d) servir como interface de usuário ao escalonamento de *jobs* em um ambiente local de processamento distribuído (*cluster*)

12|

Considere as afirmativas abaixo sobre os sistemas de arquivos virtuais:

- I. Fornece suporte a sistemas de arquivos remotos.
- II. Possui uma interface superior com os arquivos do sistema.
- III. Sua ideia principal é abstrair a parte comum aos diversos sistemas de arquivo.
- IV. Tenta integrar diferentes sistemas de arquivos em uma estrutura ordenada.

Assinale a opção que contém as afirmativas corretas:

- a) I e II
- b) I e IV
- c) II e III
- d) III e IV

13|

Uma maneira de garantir a segurança de um sistema é proteger os dados e os serviços que ele oferece contra ameaças. Dentre os quatro tipos de ameaças à segurança: interceptação, interrupção, modificação e fabricação, a única que está definida corretamente é:

- a) interrupção se refere à situação em que os dados são modificados
- b) interceptação se refere à situação em que o serviço ou dado se torna indisponível
- c) modificação se refere à situação em que uma pessoa não autorizada ganha acesso aos serviços ou dados
- d) fabricação se refere à situação em que dados adicionais são gerados, na qual normalmente esses dados não existiriam

14|

As aplicações de dados intensivos para *grids* ocorrem dentro dos domínios científicos e empresariais. Os dois ambientes necessitam prover mecanismos para atender às múltiplas exigências de acesso de usuários e das instituições envolvidas.

Para atender a essas exigências, o ambiente de *grid* deve possuir, dentre outras, as seguintes habilidades:

- a) fazer cópias locais de segurança dos dados e transferência de dados, visando à otimização de recursos da rede
- b) respostas dinâmicas às mudanças no perfil de acesso aos bancos de dados e descoberta de recursos locais ociosos
- c) publicação e compartilhamento de dados e disponibilização de métodos de descoberta de serviços baseadas no ambiente local
- d) integração de múltiplas fontes de dados independentes, distribuídas e heterogêneas, e provimento de mecanismos de acesso e/ou a replicação de dados, visando à redução do tráfego de rede

15|

Ao medir o desempenho de certo sistema, verificou-se que este passava muito tempo com a CPU ociosa e tinha um alto volume de acessos a disco.

Nesse caso, a solução para melhorar o desempenho desse sistema é:

- a) usar memória cache
- b) trocar a CPU por uma mais rápida
- c) aumentar a capacidade de memória do sistema
- d) aumentar a capacidade de armazenamento do disco

16|

O administrador do sistema precisa fazer um *script shell* (para *bash*) que receba como argumento o nome de um usuário e informe apenas se o usuário em questão está ou não logado no sistema. Não devem ser exibidas na tela outras informações adicionais sobre o usuário.

O *script* que faz esse procedimento é:

- a) if who | grep \$1; then echo \$1 logado; else echo \$1 nao logado; fi
- b) if who | grep \$1 >/dev/null; then echo \$1 logado; else echo \$1 nao logado; fi
- c) if who \$user >/dev/null; then echo \$user logado; else echo \$user nao logado; fi
- d) if who -user \$1 >/dev/null; then echo \$user logado; else echo \$user nao logado; fi

17|

A característica de autenticação simples (*single sign-on*) em *grids* permite que o usuário se autentique uma só vez e crie uma credencial por *proxy*, que pode ser usada por uma aplicação para utilizar qualquer recurso remoto em nome do usuário, normalmente por um tempo definido, sem que sejam necessárias autenticações individuais a cada recurso.

Uma credencial por *proxy* é definida como:

- a) um registro coletivo usado para manter e suportar buscas em informações de múltiplas locações
- b) um registro local utilizado para gerenciar a coleção e disseminação de informação em uma locação particular
- c) um mecanismo para que sejam invocadas operações de serviço, baseado em SOAP ou em outros protocolos
- d) um certificado digitalmente assinado, que dá direito ao portador de efetuar operações em ambiente de *grid*, tais como a submissão de *jobs* e a transferência de arquivos

18|

Técnicas eficientes para o uso de memória, como memória virtual e *caching*, podem ser aproveitadas pelo seguinte motivo:

- a) o princípio da localidade pode ser aplicado
- b) memórias dinâmicas são mais rápidas que memórias estáticas
- c) a velocidade de acesso à memória RAM aumentou muito recentemente
- d) o espaço de armazenamento na memória RAM aumentou muito recentemente

19|

Para comparar cadeias de caracteres na linguagem C utiliza-se a seguinte função/operador:

- a) =
- b) ==
- c) strcpy
- d) strcmp

20|

A arquitetura *grid* adotou certificados X.509 para as operações de criptografia e assinatura digital. Como se sabe, por diversas razões, pode-se fazer necessário revogar um certificado digital antes da sua expiração. O padrão X.509 define uma estrutura de dados assinada digitalmente, com *time-stamping*, denominada *Certificate Revocation List* (Lista de Certificados Revogados), que identifica os certificados revogados por meio de:

- a) chave privada do certificado
- b) número serial do certificado
- c) par de chaves do certificado
- d) nome do proprietário do certificado

21|

*Threads* e processos são estruturalmente distintos, embora ambos tenham seu escalonamento feito pelo gerenciamento de processos.

A principal diferença entre eles é que:

- a) *threads* possuem contexto simplificado
- b) processos são executados mais rapidamente
- c) *threads* apenas podem ocorrer em processadores *multicore*
- d) processos apenas podem ocorrer em sistemas de grande porte

22|

O administrador do sistema precisa fazer um *script shell* (para *bash*) que recebe 3 argumentos, em ordem: o caminho de um diretório que contém apenas arquivos, uma *string* a ser buscada e o nome de um arquivo para armazenar o resultado do *script*. O *script* deve entrar no diretório informado, concatenar todos os arquivos, buscar pela *string* e anexar as linhas onde a *string* é encontrada no final do arquivo de resultado. O arquivo de resultado não pode ser apagado no começo do *script*. Ao fim, o *script* deve navegar para o diretório home do usuário.

O *script* que faz esse procedimento é:

- a) `cd $1; cat * | grep $2 > $3; cd`
- b) `cd $1; cat * | filter $2 > $3; cd`
- c) `cd $p1; cat * | grep $p2 > $p3; cd`
- d) `cd $p1; cat * | filter $p2 > $p3; cd`



23|

Um analista de sistemas foi convidado para opinar sobre por que um determinado sistema apresentava desempenho ruim. Ele observou que o sistema tinha alta taxa de acesso a disco, por demanda de entrada e saída dos processos.

A razão que pode explicar o baixo desempenho é que o SO realizava a seguinte ação:

- a) entrava em *deadlock*
- b) usava dispositivos de DMA
- c) gerenciava E/S pelo algoritmo FCFS
- d) gerenciava E/S pelo algoritmo SCAN

24|

O GT foi a solução que permitiu o estabelecimento de sistemas de *grid* interoperáveis e calcou as bases para o surgimento de ferramentas de desenvolvimento em *grid*.

Dentre os problemas para os quais o GT apresentou soluções de forma pioneira, podem ser citados os seguintes:

- a) administração de SO, segurança, gerência de rede local e gerência de recursos
- b) segurança, descoberta de informações, gerência de recursos e detecção de falhas
- c) gerência de rede local, descoberta de informações, detecção de falhas e segurança
- d) gerência de recursos, detecção de falhas, administração de SO e descoberta de informações

25|

Algoritmos de eleição são usados como mecanismos para recuperar a operabilidade de algum serviço dentro de um sistema distribuído.

Com base nessa informação, é correto afirmar que:

- a) o algoritmo de (eleição em) Anel pode ser aplicado em qualquer situação de falha
- b) o relógio de Lamport é mais eficiente para realizar a eleição por ser baseado em relações temporais
- c) os serviços providos por meio de controle central não necessitam de mecanismos de eleição por já determinarem o eleito
- d) o algoritmo de (eleição) Bullying garante a definição da eleição se o meio de comunicação for confiável e suficientemente rápido

26|

Suponha um *host* com endereço IP 192.168.0.185 e máscara de sub-rede 255.255.255.128. A sub-rede à qual o *host* pertence, o endereço de *broadcast* e a faixa de IPs disponível para os *hosts* dessa sub-rede são, respectivamente:

- |                     |               |                               |
|---------------------|---------------|-------------------------------|
| a) 192.168.0.255/25 | 192.168.0.128 | 192.168.0.128 – 192.168.0.255 |
| b) 192.168.0.28/26  | 192.168.0.255 | 192.168.0.128 – 192.168.0.255 |
| c) 192.168.0.128/25 | 192.168.0.255 | 192.168.0.129 – 192.168.0.254 |
| d) 192.168.0.128/26 | 192.168.0.128 | 192.168.0.129 – 192.168.0.254 |

27|

Escalabilidade é um dos principais objetivos para os projetistas e desenvolvedores de sistemas distribuídos. As três técnicas básicas para aumentar a escalabilidade dos sistemas são:

- a) esconder a latência; distribuição; replicação
- b) centralização; replicação; comunicação síncrona
- c) transparência; esconder a latência; centralização
- d) comunicação síncrona; distribuição; transparência

28|

O gerenciamento de processos em sistemas modernos é feito, quase sempre, com o uso de preempção de processos por meio de técnicas de compartilhamento de tempo.

A introdução de processadores com vários núcleos, nesse gerenciamento, torna-se possível pelo seguinte motivo:

- a) uso de SOs multitarefas
- b) paralelização efetiva de processos concorrentes
- c) uso de *threads* para a execução de processos concorrentes
- d) separação dos demais mecanismos de gerenciamento do SO do gerenciamento de processos

29|

Suponha que o administrador de uma rede está utilizando o seguinte prefixo para uma de suas sub-redes: 128.208.0.64/26. O endereço IP pertencente a essa sub-rede é:

- a) 128.208.0.56
- b) 128.208.0.122
- c) 128.208.0.160
- d) 128.208.0.225

30|

De acordo com o modelo "ampulheta" (*hourglass*), apresentado na arquitetura de *grid* idealizada por Foster e Kesselman, o nome da camada responsável por coordenar o uso dos diversos recursos compartilhados na *grid*, tais como escalonamento ou "*scheduling*", envolvendo o seu uso coordenado é:

- a) Conectividade (*Connectivity*)
- b) Recursos (*Resource*)
- c) Coletivo (*Collective*)
- d) Estrutura (*Fabric*)

31|

Uma das razões mais importantes para utilizar a virtualização, na década de 70, foi permitir a execução de *software* legado nos extremamente caros computadores de grande porte (*mainframes*). Com o barateamento do *hardware* e a diminuição da quantidade de SOs, a virtualização perdeu a importância. Porém, no final da década de 90, a virtualização voltou a ter bastante importância.

Com base nessas informações, considere os motivos do retorno do uso da virtualização apresentados abaixo:

- I. Para facilitar o gerenciamento, uma vez que, por meio da virtualização, é possível replicar um conteúdo dinamicamente.
- II. Para permitir que dois ou mais usuários utilizem um programa em comum no mesmo computador.
- III. Facilitar a manutenção dos computadores servidores, uma vez que, por meio da virtualização, um único computador pode ser usado para manter vários servidores.
- IV. Para replicar dados para aumentar a segurança da informação.

Assinale a opção que contém os motivos corretos.

- a) I e III
- b) I e IV
- c) II e III
- d) II e IV

32|

Um usuário digitou o valor 4 na entrada padrão, ao executar o programa em linguagem C++:

```

#include <iostream>
using namespace std;
struct N {
    int A; int B;
    N *L;
};

int main()
{
    N *A, *B, *C;
    int n;

    cin >> n;
    for (int i=0; i<n; i++)
    if (!i) {
        C = new N;
        C->A = i;
        C->B = i+1;
        C->L = new N;
        C->L->A = i+1;
        C->L->B = i+1;
        C->L->L = NULL;
    } else {
        A = C;
        B = A->L;
        while (B) {
            if (A->B + B->B <= i) {
                A->L = new N;
                A->L->A = A->A + B->A;
                A->L->B = A->B + B->B;
                A->L->L = B;
            }
            A = B;
            B = B->L;
        }
        A = C;
        while (A) {
            cout << A->A << "/" << A->B << " ";
            A = A->L;
        }
    }
}

```

O resultado que será colocado na saída padrão "cout" é:

- a) 0/1 1/2 1/3 1/2 0/1
- b) 0/1 1/3 0/1 1/3 0/1
- c) 0/1 1/3 1/2 2/3 1/1
- d) 0/1 1/2 2/3 3/4 4/5

33|

A linguagem Python possui a seguinte característica:

- a) é uma linguagem compilada
- b) exige declaração de código
- c) a tupla é um tipo mutável
- d) é orientada a objetos

34|

*Deadlock* é um problema clássico em sistemas distribuídos. Para que ele aconteça, as condições necessárias, além de “manter e esperar”, são:

- a) exclusão mútua, não preempção e espera circular
- b) comunicação síncrona, espera circular e não preempção
- c) espera circular, exclusão mútua e comunicação síncrona
- d) não preempção, comunicação síncrona e exclusão mútua

**Responda às questões de 35 a 40, considerando o sistema operacional Linux.**

35|

O comando que possibilita varrer as portas de rede abertas para um determinado IP é o:

- a) ssh
- b) ping
- c) nmap
- d) nslookup

36|

O administrador precisa programar a reinicialização de uma máquina para as 20 horas e 30 minutos e deseja que os usuários logados na mesma recebam em seus terminais a mensagem “Reinicialização agendada pelo administrador.”.

O comando a ser usado com essa finalidade é:

- a) `reboot 20:30` “Reinicialização agendada pelo administrador.”
- b) `shutdown -r 20:30` “Reinicialização agendada pelo administrador.”
- c) `shutdown -h 20:30` “Reinicialização agendada pelo administrador.”
- d) `reboot -time 20:30 -message` “Reinicialização agendada pelo administrador.”

37|

Um administrador executou o comando “ps aux” em seu terminal, obtendo a seguinte saída:

USER	PID	%CPU	%MEM	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	?	S	Feb12	0:01	/sbin/init
pesq1	15230	85.3	25.5	?	R	14:05	55:06	./simulador
pesq1	15235	0.00	20.4	?	D	14:20	25:43	./buscador
pesq1	15250	80.4	45.5	?	R	14:05	55:06	./teste
pesq1	19494	0.0	0.0	?	S	14:50	0:00	sshd: pesq1@pts/9
pesq1	19495	0.0	0.0	pts/9	S	14:50	0:00	-bash
pesq1	20339	0.0	0.0	pts/9	Z	16:20	0:00	top

Apenas as linhas e colunas relevantes para o enunciado estão sendo exibidas.

O administrador deseja encerrar dois processos dessa lista. O primeiro processo a ser encerrado é aquele que está em espera no disco e deve sofrer encerramento forçado, de forma que o programa não possa interceptar e ignorar o sinal de terminação. O segundo processo é o que está usando o maior percentual de CPU dentre os exibidos na lista e deve receber sinal de terminação que possa ser interceptado, pois foi implementada uma lógica de tratamento de sinais no programa que salva o progresso do trabalho antes da finalização.

O comando usado para realizar a terminação dos processos, conforme as especificações, é:

- a) `kill -9 15250; kill -15 20339`
- b) `kill -9 20339; kill -15 15250`
- c) `kill -9 15230; kill -15 15235`
- d) `kill -9 15235; kill -15 15230`

38|

Um operador deseja alterar a senha do usuário pesquisador1. A conta do operador não é *root*, mas possui privilégio de *sudoer*. O operador não sabe a senha atual do usuário pesquisador1.

A linha de comando que deve ser executada para realizar a alteração de senha é:

- a) `passwd pesquisador1`
- b) `su pesquisador1; passwd`
- c) `sudo passwd pesquisador1`
- d) `sudo su pesquisador1; passwd`

39|

Um administrador deseja que o *script* `/scripts/verificacao.sh` seja executado de segunda a sexta, às 22:50h.

A linha a ser adicionada no “*crontab*” é:

- a) `50 22 * * 1-5 /scripts/verificacao.sh`
- b) `* * 2-6 22 50 /scripts/verificacao.sh`
- c) `/scripts/verificacao.sh -from Mon -to Fri – at 22:50`
- d) `at 22:50 from Monday to Friday run /scripts/verificacao.sh`

40|

Um administrador deseja aplicar o seguinte esquema de permissões na pasta `/home/uerj` e todos os seus arquivos e subdiretórios (de forma recursiva):

- O dono do arquivo deve ter permissão de leitura, escrita e execução.
- Adicionar permissão de leitura e escrita para os membros do grupo ao qual pertence o dono dos arquivos, sem modificar a permissão de execução já existente em cada arquivo ou diretório para os membros do grupo.
- Revogar permissão de escrita e execução para os demais usuários.

O comando a ser usado com essa finalidade é:

- a) `chmod -R 764 /home/uerj`
- b) `chmod -R 746 /home/UERJ`
- c) `chmod -R u=rwx,g=rw,o=r /home/uerj`
- d) `chmod -R u=rwx,g+rw,o-wx /home/uerj`