



Universidade Federal Rural do Semi-Árido
Departamento de Engenharias e Tecnologia
Câmpus Pau dos Ferros



Primeira Avaliação Prática - Unidade I

Docente: Walber José Adriano Silva

Data de entrega: 11/09/2022

Disciplina: Sistemas Operacionais

1. Criando instâncias na AWS via AWS Academy (Nota: 2,0):

- Crie uma instância com a imagem do **Amazon Linux** e envie a imagem dos detalhes da instância com a informação do IPv4 privado e público no painel EC2, bem como a informação do AMI ID (disponível em EC2>Instances>ID da instância> Details> Instance Details> AMI ID);
- Acesse a instância criada no item a via protocolo SSH e instale o servidor Apache2. Configure para que o código do repositório git hub de endereço: <https://github.com/walberjose/pacman-canvas> esteja no diretório raiz do servidor web; Enviar a imagem do navegador mostrando a barra de endereço com a informação do endereço IPv4 da instância e o software instalado com sucesso.

Dica: Lembre-se de liberar a porta de entrada 80 do protocolo HTTP no security group da instância e de inicializar o serviço do servidor web Apache2 após a instalação.

Observação 01: O diretório raiz padrão de uma instalação do servidor Apache2 é o /var/www/html

Observação 02: O não envio do AMI ID da instância resultará em não pontuação da questão.

2. Interagindo com o núcleo do Sistema Operacional Linux (Nota: 2,0):

- Crie uma máquina virtual Linux no serviço AWS/EC2 (vide seção “Criação e execução de uma instância no AWS/EC2” da “Aula 03 - Apresentação do AWS Academy - UFERSA”). Uma vez logado na máquina virtual execute alguns comandos no terminal:
- No diretório home do usuário, crie um diretório chamado tmp com o comando: mkdir tmp;
- Entre no diretório tmp, com o comando: cd tmp;

- Crie um arquivo chamado “SEUNOME.c”, com o programa nano. Adicione neste código chamadas de sistema que imprimam o texto “Olá, Sistemas Operacionais. Sou uma função que imprime no terminal via modo kernel!\n”;
- Compile o arquivo com o comando: gcc SEUNOME.c
- Verifique se houve a criação do arquivo objeto *a.out* no diretório corrente através do comando ls, e caso haja algum problema de compilação, conserte.
- Envie o resultado da atividade retirando *printscreens*, ou registros fotográficos, de cada uma das seguintes ações:
 - a) Imprimir o diretório corrente com o comando *pwd*;
 - b) Extrair todas as strings do binário *a.out*, com o comando: *strings a.out*
 - c) Imprimir as chamadas de sistema e sinais do executável *a.out*, com o comando: *strace ./a.out*
 - d) Imprimir o código em Assembly do binário, com o comando: *objdump -dM intel a.out*

Observação 01: O não utilizar a função *printf* e nem a biblioteca “*stdio.h*” da linguagem de programação C, caso contrário, não será pontuada a questão. Ao invés, utilize o conjunto de bibliotecas do kernel *unistd.h*, *sys/types.h*, *sys/stat.h*, *fcntl.h* para essa impressão.

3. Criando múltiplos processos (Nota: 2,0):

- a. Crie uma instância Linux Ubuntu no serviço EC2 da AWS. *Envie a imagem dos detalhes da instância com a informação do IPv4 privado e público no painel EC2, bem como a informação do AMI ID;*
- b. Instale o utilitário *cowsay* (<https://en.wikipedia.org/wiki/Cowsay>) na sua instância. Demonstre que o utilitário está devidamente instalado executando o com comando:
echo "Olá, SEUNOME" | cowsay -f dragon-and-cow
- c. Crie e programe em um arquivo chamado “SEUNOME-cowsay.c” onde esse programa, quando executado, irá gerar três processos onde: 1) *cowsay*; 2) *cowsay*; 3) *date*. Deverá ser utilizado a chamada de criação de novos processos *fork()* e o executor de programas *execve()*. *Enviar o programa fonte SEUNOME-cowsay.c;*
- d. Demonstre que seu programa em C executa o que foi solicitado no *item c*. *Enviar a execução do programa na sequência em que um novo processo irá gerar o cowsay com a mensagem “Primeira”, depois outro novo processo gerando o cowsay com a mensagem “Segunda” e, por fim, o comando date, demonstrando a data corrente.*

```

wãlber@guerreiro:~/tmp/tmp/tmp/tmp/tmp$ ./a.out
Oi, sou o processo 356003
[retval: 356004] sou 356003, filho de 299816
[retval: 0] sou 356004, filho de 356003

< Primeira! >
-----
      ^ ^
      (oo)\_____
      (__) \       )\\/
           ||-----w||
           ||         ||

< Segunda! >
-----
      ^ ^
      (oo)\_____
      (__) \       )\\/
           ||-----w||
           ||         ||

seg 05 set 2022 14:05:10 -03

```

Observação 01: Para o processo pai esperar o término do processo filho, é possível utilizar a função `waitpid`, por exemplo, `waitpid(retval, &status, 0)`;

Observação 02: Para mudar a figura do cowsay, você pode utilizar qualquer um dos arquivos do utilitário listados no comando “`cowsay -l`”

Dica: Para passagem de parâmetros ao `execve`, crie um arranjo aos moldes de: <https://stackoverflow.com/questions/30149779/c-execve-parameters-spawn-a-shell-example>

4. Criando contêineres (Nota: 2,0):

- Crie uma instância **Linux Red Hat Enterprise Linux** no serviço EC2 da AWS. *Envie a imagem dos detalhes da instância com a informação do IPv4 privado e público no painel EC2, bem como a informação do AMI ID;*
- Instale o motor Docker na instância. *Demonstre o funcionamento de contêineres, na sua instância, executando e enviando ao menos duas execuções de contêineres na lista: <https://hub.docker.com/r/wernight/funbox>*

Observação 01: Utilize a nomenclatura b.1, b.2, etc. para demonstrar as imagens da execução dos contêineres.

5. Aninhamento de virtualização (Nota: 2,0):

Contexto: Uma atividade chegou para você com um desejo muito peculiar de um cliente. O cliente afirma que necessita ter o Sistema Operacional Linux Debian 11 instalado e virtualizado dentro de uma instância AWS EC2 com imagem do Sistema Operacional Microsoft Windows 2022. Segundo o cliente, este requisito de virtualização aninhada se deve a um software legado que utiliza esses dois sistemas operacionais em conjunto para poder operar. Apesar de você informar ao cliente que há outras alternativas eficientes e eficazes para endereçar o requisito deste software legado, o cliente insistiu para que seja realizado dessa forma. Dessa

forma, realize a instalação do Debian 11 dentro de uma instância EC2 que executa o Windows 2022.

Objetivo: instalar o Debian 11 dentro de uma instância EC2 que executa o Windows 2022.

- a. (Nota 0,5) Demonstre que não é possível realizar a instalação do [VirtualBox](#) (ou [VMWare](#)) no Windows 2022 em instâncias EC2 que estão executando na modalidade de preço Sob demanda. Enviar três imagens para este item:
 - i. *Envie a imagem dos detalhes da instância com informação do DNS IPv4 privado no painel EC2;*
 - ii. *Envie a imagem da execução do comando ipconfig no terminal cmd.exe do Windows 2022. Deve estar visível que é o mesmo endereço IPv4 privado reportado no painel EC2 para a sua instância;*
 - iii. *Envie uma imagem da impossibilidade de instalar uma máquina virtual Debian 11 no VirtualBox dentro da instância EC2 sob demanda Windows 2022. Dica: Qual erro aparece quando se tenta instalar o Sistema Operacional [Debian 11](#) dentro de uma instância Sob demanda Microsoft Windows 2022 utilizando um virtualizador?*
- b. (Nota 0,5) Forneça uma justificativa técnica sobre o porquê da virtualização com o VirtualBox (ou VMWare) não poder atender o pedido do cliente.
- c. (Nota 1,0) Demonstre que é possível atender o pedido do cliente, com outro tipo de virtualização/emulação, e instale um virtualizador/emulador que permita a instalação de um Sistema Operacional [Debian 11](#) dentro de uma instância sob demanda Microsoft Windows 2022. Após a instalação *envie uma imagem de um usuário logado no Debian 11, e que apareça a informação do IPv4 privado da instância Windows 2022 solicitada no item a.*

Dica:

- I. Para executar o cmd.exe vá no Windows -> Run e digite o cmd.exe
- II. Caso o desempenho da sua instância Windows 2022 esteja muito degradado e lento, utilize um [Tipo de Instância](#) que consuma mais créditos.
- III. Observe qual tipo de ISO Debian você irá instalar na máquina, se será i386 - arquitetura 32 bits ou amd64 - arquitetura 64 bits.