

## Mãos à obra: Salvando os arquivos com extensão .log

No nosso script, nós fomos capazes de obter a data e horário e salvamos dentro do arquivo com o nome do processo **.log**. Agora iremos realizar a última etapa que seria de salvar o tamanho da memória alocada para cada processo dentro desse arquivo.

Abra novamente o script que estamos trabalhando com o editor de texto de sua preferência. Feito isso, iremos para cada número do processo obter o tamanho da memória alocada e iremos guardar o resultado em uma variável chamada de **tamanho\_processo**

```
tamanho_processo=$(ps -p $pid -o size | grep [0-9])
```

Esse valor apresentado é em kB, vamos dividir esse valor por 1.024 para que possamos obter assim o valor em MB e salvar o resultado no arquivo com a extensão **.log**

```
echo "$(bc <<< "scale=2;$tamanho_processo/1024") MB" >> $nome_processo.log
```

Para finalizar, vamos guardar todos os arquivos dentro de um diretório chamado **log** que estará dentro do diretório **Scripts**. Será que esse diretório existe? Não sabemos, vamos pedir para que nosso script faça essa verificação. Logo abaixo da linha com o interpretador coloque:

```
if [ ! -d log ]
then
    mkdir log
fi
```

Feito isso, vá até as linhas com os nomes dos arquivos com a extensão **.log** e coloque que esses arquivos serão salvos nesse diretório **log**, ficando:

```
echo -n $(date +%F,%H:%M:%S,) >> log/$nome_processo.log
...
echo "$(bc <<< "scale=2;$tamanho_processo/1024") MB" >> log/$nome_processo.log
```

Por fim, envolva todo o código em uma função chamada de **processos\_memoria** e faça a chamada dessa função. Ao chamar a função, verifique o status de saída e imprima uma mensagem no terminal.

```
processos_memoria
if [ $? -eq 0 ]
then
    echo "Os arquivos foram salvos com sucesso"
else
    echo "Houve um problema na hora de salvar os arquivos"
fi
```

Agora execute o seu script, qual o resultado?

