**Universidade do Estado de Santa Catarina – Bacharelado em Ciência da Computação**
**TE em Ciência da Computação – Prof. Rafael R. Obelheiro**

# Exercícios – Objetivos 107.2 e 108.2

**Objetivo 108.2** Logs: `rsyslogd`, `systemd-journal`, `journalctl`

**Objetivo 107.2** Temporizadores do `systemd`

## Objetivo 108.2

1. Existem pelo menos três formas de logar no seu Ubuntu Server:

   i. Via interface gráfica;

   ii. Via console;

   iii. Via SSH (executando `ssh torvalds@localhost` uma vez logado por um dos caminhos anteriores).

   Efetue login por esses três caminhos, certificando-se de, para todos eles, informar uma senha inválida antes da correta. Depois, use os comandos `last`, `sudo lastb` e `sudo faillog -a` para verificar se as tentativas bem e malsucedidas de login são devidamente registradas.

2. Use o comando `lastlog` para descobrir os usuários:

   (a) que fizeram login nos últimos 3 dias;

   (b) que já fizeram login, mas não nos últimos 2 dias.

   É possível que nenhum usuário no seu Ubuntu Server satisfaça alguma das condições acima. Nesse caso, você pode testar o comando no Linux oficial do lab.

3. Considere o arquivo `rsyslog.conf` abaixo:

```
auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          /var/log/syslog
daemon.*                        /var/log/daemon.log
mail.*                          /var/log/mail.log
mail.warn                       /var/log/mail.warn
local0.*                        /var/log/local0.log
*.=crit                         /var/log/crit.log
*.err                           /var/log/err.log
```

   Especifique em quais arquivos seriam registrados logs com as prioridades (*facility.level*) abaixo:

   (a) `daemon.info`

   (b) `auth.notice`

   (c) `authpriv.crit`

   (d) `mail.emerg`

   (e) `local0.info`

   (f) `local1.err`

4. Considere que seu sistema realiza um backup diário por meio de um script `bkp-sys` que é executado via cron. Esse script deseja gerar mensagens de log no início e no final do backup, conforme o exemplo abaixo:

```
Aug 18 00:01:00 linux51 bkp-sys[923]: Iniciando backup do sistema
Aug 18 00:02:45 linux51 bkp-sys[923]: Backup finalizado
```

   Nesse exemplo, os *timestamps* correspondem aos instantes em que as mensagens foram geradas, `linux51` é o nome da máquina, e 923 o PID do script.

   Mostre como o comando `logger` poderia ser usado para gerar essas mensagens de log, sabendo que elas devem usar o recurso (*facility*) `local3` e o nível de prioridade `info`.

5. Para configurar um servidor de logs, edite o arquivo `/etc/rsyslog.conf`, e descomente as linhas abaixo:

```
#module(load="imudp")
#input(type="imudp" port="514")
```

   A seguir, crie o arquivo `/etc/rsyslog.d/05-remote.conf`, com o seguinte conteúdo:

```
$template RemoteLog, "/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"
if ($hostname != $fromhost) then {
  *.* ?RemoteLog
  & stop
}
```

   Crie o diretório `/var/log/remote`, com dono `syslog`. Com essa configuração, logs remotos serão armazenados no diretório `/var/log/remote`, separados por máquina e programa.

   Por fim, reinicie o serviço `rsyslog` do `systemd`.

6. Peça para um(a) colega criar o arquivo `/etc/rsyslog.d/99-loghost.conf`, com o seguinte conteúdo:

```
local7.*     @IPaddr
```

   onde `IPaddr` é o endereço IP da sua máquina. A seguir, peça para que ele(a) reinicie o `rsyslog` e teste a configuração executando o comando

```
$ logger -p local7.info Teste log remoto
```

   Verifique se esse log é armazenado sob `/var/log/remote` na sua máquina.

7. No seu sistema, quantos arquivos de log possuem frequência de rotação diária, semanal, mensal e anual?

8. Descreva a política de rotação (frequência, quantas versões anteriores são mantidas, uso de compressão) para os logs abaixo no seu sistema:

   - `/var/log/auth.log`
   - `/var/log/ubuntu-advantage-timer.log`
   - `/var/log/dmesg`

9. Ajuste a configuração do seu sistema para que a política *default* seja rotacionar logs diariamente, mantendo logs dos últimos 14 dias. Os arquivos de log devem ser comprimidos, com exceção do mais recente (`.1`).

## Objetivo 107.2

10. Use um temporizador transiente do `systemd` para executar o script abaixo a cada 15 minutos.

```
#! /bin/sh

HOME=/home/torvalds
/usr/bin/w >> ${HOME}/w.log
```

11. Salve o script abaixo como `df-ext4.sh`, e ajuste suas permissões de execução.

```
#! /bin/sh

HOME=/home/torvalds
logf=${HOME}/df-ext4.log
# redireciona stdout e stderr para o arquivo de log
exec >>${logf}
exec 2>&1

echo "----------------------------"
/bin/date
echo
/bin/df -h -t ext4
echo
```

    Execute o script, e verifique que a saída aparece no arquivo de log.

    A seguir, salve o arquivo abaixo como `df-ext4.service`, e copie-o para o diretório apropriado. Habilite-o para uso (`systemctl enable`), verifique que ele está habilitado (`systemctl status`), e inicie-o (`systemctl start`), verificando que o script é executado.

```
[Unit]
Description=unidade de servico para df-ext4

[Service]
ExecStart=/bin/bash /home/torvalds/df-ext4.sh
User=torvalds

[Install]
WantedBy=multi-user.target
```

    Por fim, crie um temporizador `systemd` para executar o script todos os dias às 05:00 e às 23:00. **DICA**: teste seu temporizador programando a execução para daqui a 1–2 minutos.

12. Qual a diferença entre executar um comando usando `systemd-run --on-activesec=1m` e criar um temporizador `systemd` com atributo `OnActiveSec=1m` que execute o mesmo comando em uma unidade de serviço?