Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC
Professor: Reinaldo Gomes
Disciplina: Administração de Sistemas

Prática 06 – systemd

■ O systemd

O systemd init é uma das implementações mais recentes do init do Linux. Além de gerenciar o processo normal de boot, o systemd tem como objetivo incorporar vários serviços-padrão do Unix, como cron e inetd.
Quando o systemd executa no momento do boot ele irá:

- Carregar suas configurações
- Determinar o objetido do boot, que normalmente se chama default.target
- Determinar todas as dependências do objetivo de boot default, as dependências dessas dependências e assim por diante até que todas sejam resolvidas
- Ativar as dependências e o objetivo do boot
- Após o boot, o systemd pode reagir aos eventos do sistema e ativar componentes adicionais

Um dos aspectos mais interessantes do systemd é que ele não lida apenas com processos e serviços, ele também pode montar sistemas de arquivos, monitorar sockets de rede, etc. Cada tipo ação que pode ser realizada é chamada de tipo de unidade (unit type) e cada capacidade específica é chamada de unidade (unit).
Para que um sistema Unix realize o seu processo de inicialização, os principais tipos de unidade utilizados são:

- Unidades de serviço (service units) - controlam os daemons de serviços em um sistema Unix
- Unidades de montagem (mount units) - controlam a associação de sistemas de arquivos ao sistema
- Unidades de algo (target units) - controlam outras unidades

Tipicamente o boot default é uma unidade de algo que agrupa várias unidades de serviços e de montagem com dependências. Como resultado é fácil ter uma visão do que irá acontecer no boot, sendo possível até mesmo criar um diagrama de árvore de dependência usando o comando

```
$ systemd-analyze dot --order | dot -Tsvg > systemd-user.svg
```

Você pode perceber que a árvore é bastante extensa uma vez que uma grande quantidade de unidades são executadas por padrão no momento da inicialização do sistema.

Uma série de outros comandos podem ser usados para realizar o gerenciamento dos serviços no Linux. Nessa prática vamos ver algumas das principais operações que podemos realizar.

■ Como criar um novo serviço com systemd

Inicialmente precisamos criar um script chamado “novo_servico.sh” no diretório “/usr/bin/” para a partir dele criar um novo serviço no sistema. Após criado o script deve ter o conteúdo a seguir:

```
#!/bin/bash

DATE=$(date '+%H:%M:%S %d-%m-%Y')

echo "Novo servico iniciado em " $DATE

while :
do
echo "Monitorando..."
sleep 10;
done
```

Após criado mude as permissões do arquivo para que ele tenha permissão de execução.

Uma vez definido o novo programa que deve rodar no nosso serviço é necessário realizar a indicação para o systemd que ele se trata de um novo serviço. Para tanto devemos criar um arquivo de Unidade no diretório "/etc/systemd/system" chamado novo_servico.service. O arquivo deve possuir o seguinte conteúdo:

```
[Unit]
Description=Novo servico de exemplo.

[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/novo_servico.sh

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

Esse arquivo faz a definição de um serviço simples e a parte crítica nele é a diretiva **ExecStart**, que é responsável por definir qual comando será usado para iniciar o serviço.

Depois de criado o arquivo mude as permissões dele para que ele tenha permissão de leitura e escrita para seu dono, e leitura para os demais.

Seu programa já está definido como um novo serviço do sistema.

Para iniciar um serviço do systemd, executando instruções no arquivo de unidade do serviço, use o prefixo start no comando. Se você estiver executando como um usuário não-root, você terá que usar o sudo uma vez que este irá afetar o estado do sistema operacional:

```
$ sudo systemctl start novo_servico.service
```

Como o systemd sabe a localização dos arquivos de descrição dos serviços (*.service) você pode também executar os comandos de gerenciamento de serviços da seguinte forma:

```
$ sudo systemctl start novo_servico
```

Para parar um serviço em execução, você pode usar o comando prefixo stop no comando:

```
$ sudo systemctl stop novo_servico.service
```

Esses comando são úteis para iniciar ou parar serviços durante a sessão atual, mas em muitos casos desejamos configurar um serviço para que ele seja iniciado automaticamente no boot do sistema. Para tanto, é necessário habilitá-lo na inicialização através da opção enable:

```
$ sudo systemctl enable novo_servico.service
```

Isto irá criar um link simbólico do arquivo de serviço para o local no disco onde systemd procura por arquivos de inicialização automática (geralmente /etc/systemd/system/some_target.target.wants).

Para desativar o serviço que é iniciado automaticamente, você pode usar o disable:

```
$ sudo systemctl enable novo_servico.service
```

Isto irá remover o link simbólico que indica que o serviço deve ser iniciado automaticamente.

É importante reforçar que habilitar um serviço com o comando enable não irá iniciá-lo na sessão atual. Se você deseja iniciar o serviço e habilitá-lo no boot, você terá que usar os comandos start e enable.

■ Verificando o Status de Serviços com o Systemctl

Já para verificar o status de um serviço em seu sistema, você pode usar o comando status:

```
$ sudo systemctl status novo_servico.service
```

Isto irá fornecer-lhe com o estado do serviço, a hierarquia cgroup, e as primeiras linhas de log.

Existem também métodos para a verificação de estados específicos. Por exemplo, para verificar se a unidade está ativa (em execução), você pode usar o comando is-active:

```
$ sudo systemctl is-active novo_servico.service
```

Isto irá lhe retornar o estado da unidade atual, que é geralmente active ou inactive. O código de saída será “0” se ele estiver ativo, tornando o resultado mais simples para analisar programaticamente.
Para ver se a unidade for ativada, você pode usar o comando is-enabled:

```
$ sudo systemctl is-enabled novo_servico.service
```

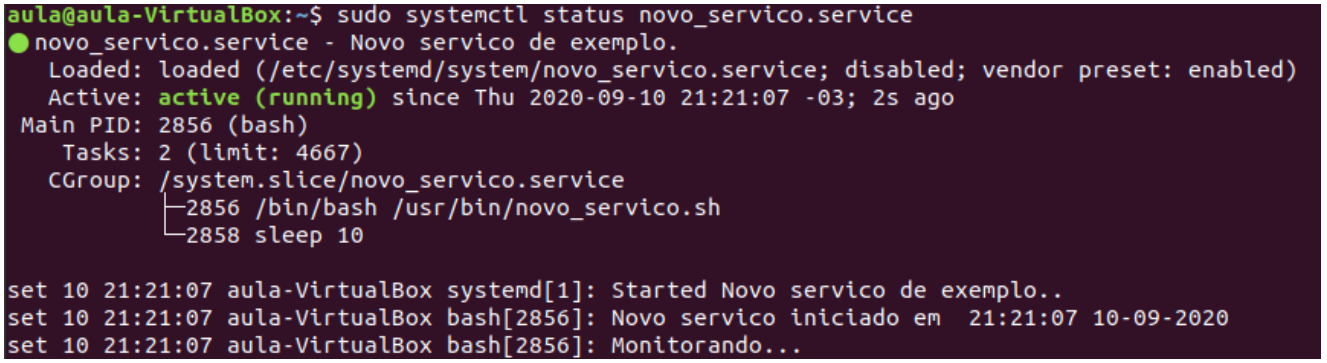
A saída será se o serviço está enabled ou disabled e voltará a definir o código de saída para “0” ou “1”, dependendo da resposta à pergunta do comando.

Uma terceira verificação é se a unidade está em um estado de falha. Esta indicará se existe um problema ao iniciar a unidade em questão:

```
$ sudo systemctl is-failed novo_servico.service
```

Isso irá retornar active se estiver funcionando corretamente ou failed se ocorreu um erro. Se a unidade foi intencionalmente parada, ele pode retornar unknow ou inactive. Um status de saída “0” indica que ocorreu uma falha e um status de saída “1” indica qualquer outro status.

Para o novo serviço que criamos, se ele for iniciado corretamente você deve ter esse resultado ao verificar o seu status:



■ Reiniciando e Recarregando com o systemctl

Para reiniciar um serviço em execução, você pode usar o comando restart:

```
$ sudo systemctl restart novo_servico.service
```

Se a aplicação em questão é capaz de recarregar seus arquivos de configuração (sem reiniciar), você pode usar o comando reload para realizar esse processo sem que o serviço tenha que ser parado:

```
$ sudo systemctl reload novo_servico.service
```

Se você não tem certeza se o serviço tem a funcionalidade para recarregar sua configuração, você pode usar o comando reload-or-restart. Isto irá recarregar a configuração no local, se disponível. Caso contrário, ele irá reiniciar o serviço para que a nova configuração tome efeito:

```
$ sudo systemctl reload-or-restart novo_servico.service
```