Trabalho Prático 2

Douglas W Lopes e Lucas Pereira Monteiro

11 de Novembro 2015

1 Introdução

SNMP acrônomo para Simple Network Management Protocol, ou em português: Protocolo Simples de gerenciamento de redes. É um protocolo da camada de aplicação para transporte de informações referentes ao gerenciamento de dispositivos conectados em uma rede.

Sua aplicação atende a necessidades de coletar e analisar informações de uma rede, pois para um gerente de redes é sempre importante saber sobre os disponsitivos de tal rede. Segundo, as informações do link : http://www.ti-redes.com/gerenciamento/snmp/intro/, acessado em 2 de dezembro de 2015, O SNMP "possibilita que administradores de rede gerenciem o desempenho da uma rede monitorando interfaces, processadores, memórias de equipamentos como roteadores, switches, dispositivos wireless e servidores."

Logo abaixo, temos uma aplicação do portocolo SNMP, para o gerenciamento de rede NMS-Network-Management Systems - Sistema responsável pelo monitoramente e controle dos dispositivos gerenciados. Permiti que os administradores de redes visualizem as informações de leitura SNMP, seja por meio de gráfico, tabelas, relatórios, alertas por email ou envio de sms. Como exemplos de NMS podemos citar: MRTG, Cacti, Nagios, PRTG, CiscoWorks entre outros. Esse acompanhamento é muito importante para manter a rede em bom funionamento, permitindo que uma rede possa trabalhar com vários dispositivos diferentes que o "gerente" saiba,determine e modifique essas informações e/ou equipamentos da rede.

Nesse trabalho em específico utilizaremos o protocolo de maneira mais simples utilizando o enderaçamento IPV6 e UDP (geralmente já utilizado quando falamos de SNMP). A intenção é recuperar informações das máquinas conectadas a redes, como por exemplo: Nome da máquina, endereço IPV4, IPV6 e etc. No trabalho o programa agente se conectará com o programa monitor, para troca dessas informações/*Adicionar quem vai ceder e quem vai receber essas inofrmaçõs*/; o monitor será capaz de receber esses vários agentes e gerenciar todos os requisitos de informações feitas à ele.

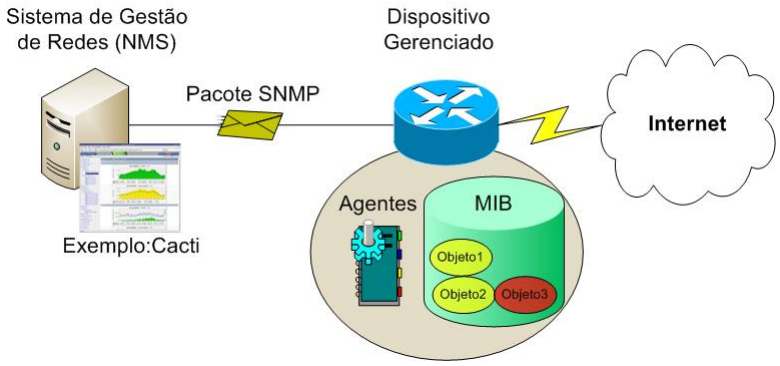

Figure 1: Representação de um sistema NMS para gereciamento de redes utilizando o protocolo SNMP **Fonte:** Google

2 Solução do Problema

Esse último trabalho prático da disciplina, foi relativamente menos complexo que os outros. E também estamos numa etapa que exige a repetição de algumas estruturas, porém só mudamos alguns detalhes, como por exemplo: agora usaremos o protocolo não orientado UDP, visualizaremos o IPV6, entre outros.

Buscamos nesse trabalho uma resolução simples e objetiva do problema, temos apenas 3 estruturas de dados, que em conjunto conseguem solucionar o problema. Logo abaixo entramos mais no detalhe de cada estrutura, mas antes vamos especificar um pouco mais os papéis de Monitor e Agentes nesse trabalho prático.

2.1 Monitor

O nosso monitor agora, análogo ao servidor dos últimos trabalhos, possui algumas diferenças. Nesse momento estamos trabalhando com o UDP, fazendo troca de datagramas. Além de utilizar o protocolo IPV6 para a troca desses datagramas. A partir da criação do socket e dado início a comunicação passiva, agora aguardaremos conexões de agentes; e por fim esses farão solicitações de informações,já pŕe definidas pela expecificação do Trabalho, da máquina onde o monitor está instalado.

Feito o pedido, imprimiremos na tela do agente tais informações. Abaixo será explicitado como ocorre essa coleta de informação, estruturas usadas e funções para coleta dessas informações.

2.2 Agente

O agente, análogo ao cliente, inicia a comunicação ativa. Ele fará a requisisão e aguardará pela impressão em sua tela dos dados requisitados.

3 Estrutura de Dados

3.1 Utilidades

Essa é a estrututura de dados que possuí a parte principal desse trabalho prático. Mais especificamente, declaramos aqui quais serão os comandos, como eles serão lidos e como serão colocados no Buffer para a impressão no cliente. Segue os principais serviços e utilização.

GetInfo Basicamente o que fazemos nessa estrutura repetidamente, é pela utilização do FILE * fp = popen(), que exucuta os comandos prévios declarados na estrutura e armazena-os diretamente no buffer. Logo temos 5 comandos fp = popen() para executar as requisições. Uma requisão receberá 5 informações como endereço IPV4, Espaço em disco, etc. Para cada um desses foi criado um "comando", que esse fp=popen executará no momento em que o serviço de requisão for solicitado pelo agente. Assim a cada fp = popen(), armazerá um dado requisitado e em seguida já o colocamos diretamente no buffer, e em seguida limpamos o buffer de lixo, e já o deixamos pronto, para quando chegar ao cliente, sua impressão seja "mais agradável" na tela do agente.

Declarações Em utilidade declaramos comandos e os relacionamemos com cada tipo de retorno, para que possamos executar o fp = popen(); porém isso é bem específico de C, e estamos detalhando mais aqui para entendimento do monitor e professores na hora da correção.

beautifyBuffer Esse serviço, trata de limpar o buffer de lixos e caracteres "estranhos", após cada requisição, como IPV4, IPV6, etc;

4 Algumas decisões de implementação

A única decisão de implementação de grande interesse, foi o fato do não uso do comando System. Pois decidimos usar o fp = popen(), que ao invés de manipular arquivos, simplesmente declaramos e criamos comandos para coletar as informações que queremos e podemos copíar e colar diretamente já no buffer. E deixar no agente a tarefa de exibir todas essas informações. Porém bem como no System o fp = popen() trabalha da mesma maneira e realizamos essa função no monitor.

5 Testes

Os testes possuem algumas adições de prints para comprovar o funcionamento correto e entendimento por nossa parte. Levando a simplicidade das saídas, realizamos um teste, já que não há grandes modificações ou exigências de execuções mais específicas, por parte da especificação.

Frizamos novamente que esses testes tem fins didáticos e que nossa intenção é facilitar o entendimento do monitor em relação ao nosso projeto.Quaisquer eventuais dúvidas serão esclarecidas no momento da entrevista. Caso seja necessário podemos demostrar em alguma entrevista caso haja quaisquer dúvidas.

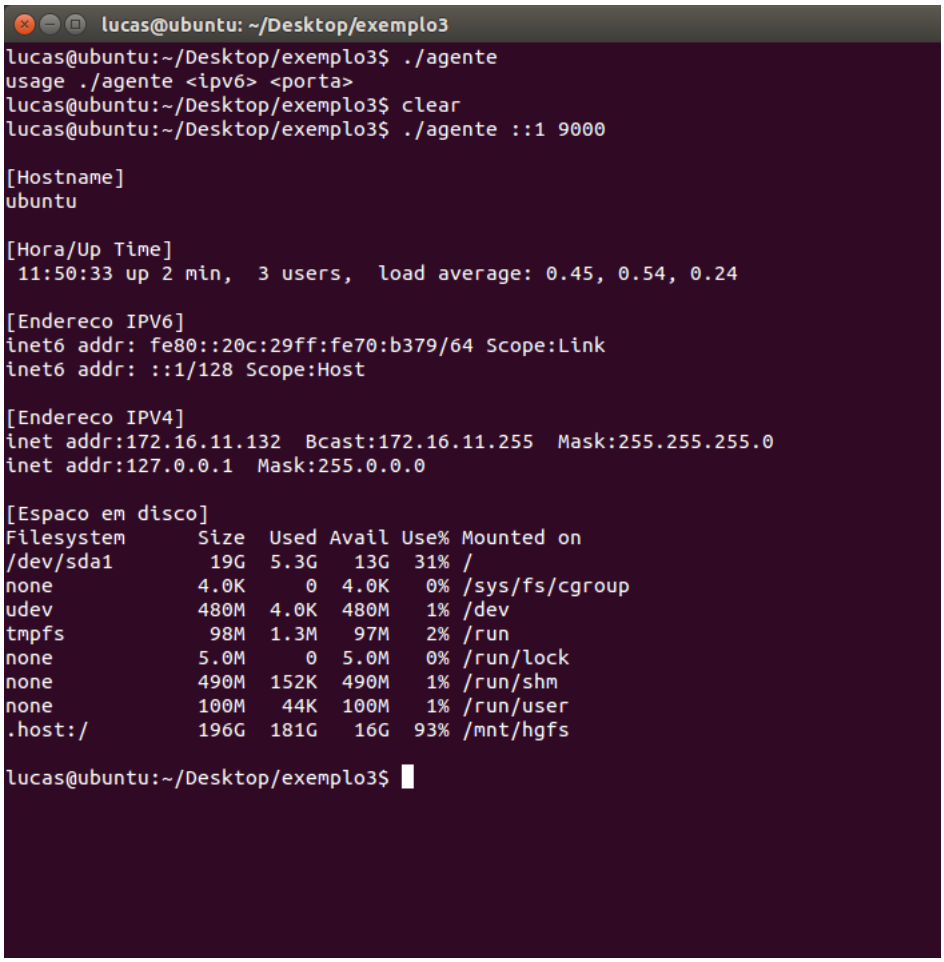

Figure 2: Representação das informações impressas para o agente, oriundas do monitor **Fonte:** Trabalho Prático 2

6 Conclusões

Nesse último trabalho prático tivemos a oportunidade de trabalhar com IPV6, UDP entre outras novas estruturas. Essa mudança foi satisfatória uma vez que podemos colocar em prática ainda mais sobre os conhecimentos da displina, todos: desde de TCP, IPV4 até UDP, IPV6.

Além de tudo, tivemos oportunidade de no final da displina revisar os conceitos estudados durante todo o semestre, além de colocar em prática o que estamos estudando em camada de redes, transporte e aplicação; que está bem próximo desde trabalho. Apesar de um trabalho simples, houve grandes desafios, como colocar as informações requistadas no buffer, re-lembrar sobre ar-

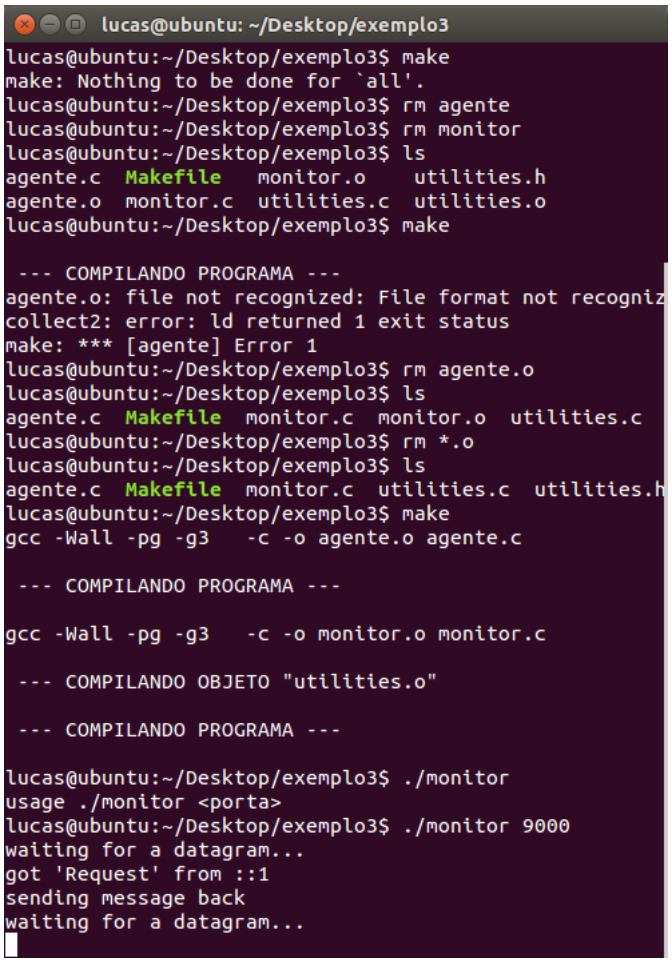

Figure 3: Representação do funionamento correto do monitor, com delaração de portas e esperando por datagramas **Fonte:** Trabalho Prático 2

quivos, pois inicialmente nossa ideia era de usar o System(), além de lidar com protocolos IPV6 e UDP. Mas mesmo com todos esses desafios foi um trabalho gratificante de se fazer e muito interessante, dando o gosto de terminar a displina com chave de ouro e podendo ter a certeza que todo o conteúdo lecionado foi bem ensinado, tivemos a oportunidade de colocar em prática, além de todo o conhecimento que levaremos para a vida profissional em redes.

Obrigado por esse semestre,

7 Referências

N.Ziviani,*Projeto de Algoritmos com implementaçao em C e Pascal*,Cengage Learning,3ªEd Revista e Ampliada,2011.

T.H.Cormen,E.Leiserson,R.L.Rivest,C.Stein, *Algoritmos:Teoria e Prática*,Elsevier,Traduç ao da 3ªEdiç ao Americana,2012.

Sockets Tutorial:http://www.linuxhowtos.org/C-C++/socket.html-acessado pela última vez para esse trabalho em 05/10/2015

Materiais disponibilizados no moodle: A especificação do TP, exemplos de sockets e bilbiotecas de socket.

http://www.ti-redes.com/gerenciamento/snmp/intro/

http://www.4linux.com.br/o-que-e-snmp