ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DE LEIRIA
INSTITUTO POLITÉCNICO DE LEIRIA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Sistemas Distribuídos e Paralelos

2005/2006

Engenharia Informática

2.º ANO regime Diurno 3.º ANO regime Nocturno

Engenharia Informática e Comunicações

2.º ANO

Securitas

Relatório da implementação do serviço *Securitas* do sistema de *Buscas Distribuídas*.

Documento elaborado por:
Sérgio Miguel Neves Lopes – aluno n.º 10635

Versão 1.0

Novembro/2005

ÍNDICE

1 Introdução

1.1 Descrição Geral do Trabalho

O BUD – Buscas Distribuídas, pretende ser um sistema distribuído com a capacidade para, regularmente, efectuar a pesquisa de ficheiros no parque informático de uma instituição.

1.1.1 Descrição da Primeira Etapa – Implementação do *Securitas*

Esta primeira etapa tem por objectivo a implementação do serviço de autenticação e autorização denominado *Securitas*.
O objectivo é a implementação de um servidor iterativo e *stateless* que permita a autenticação de utilizadores, autorização de acesso a recursos e gestão de sessões de utilizadores.

1.2 Definições e Acrónimos

Os acrónimos importantes a reter para uma melhor compreensão do documento são:

ACRÓNIMO	DESIGNAÇÃO
IP	Internet Protocol

2 Descrição Geral das Principais Funções Implementadas

2.1 Descrição de “parse_maquinas”

A função tem por objectivo ler o ficheiro que contém os IPs, os utilizadores e as pastas a que estes têm acesso. Ao percorrer o ficheiro indicado a função tenta identifica linhas inválidas continuando a execução para as linhas seguintes caso encontre alguma linha que não esteja de acordo com o formato esperado para o ficheiro.

A função não admite informação útil após um cardinal, “#”, sendo este usado para indicar um comentário, assim para a linha “192.168.0.1;ei10635;c:\xpto|d:\xpto#123”, a função considera “192.168.0.1” como o IP da máquina onde se encontra o serviço de buscas instalado, “ei10635” como “login” de utilizador e “c:\xpto” e “d:\xpto” como o conjunto de pastas a que o utilizador tem acesso, ignorando “123” por considerar um comentário. Deste modo é possível introduzir comentários após informações úteis.

Tal como no formato do ficheiro de contas, o “ponto-e-vírgula”, não é um símbolo válido para o *login* do utilizador.

Caso sejam incluídos nomes de máquinas no ficheiro, em vez dos IPs, a função tenta efectuar a sua resolução de forma a obter o(s) IP(s) da máquina.

2.2 Descrição de “parse_contas”

A função tem por objectivo ler o ficheiro que contém os utilizadores e as palavras-chave. Ao percorrer o ficheiro indicado a função tenta identifica linhas inválidas continuando a execução para as linhas seguintes caso encontre alguma linha que não esteja de acordo com o formato esperado para o ficheiro.

A função não admite informação útil após um cardinal, “#”, sendo este usado para indicar um comentário, assim para a linha “ei10635;xpto#123”, a função considera “ei10635” como *login* de utilizador e “xpto” como palavra-chave ignorando “123” por considerar um comentário. Deste modo é possível introduzir comentários após informações úteis.

Embora o “ponto-e-vírgula” seja um símbolo delimitador, ele é permitido nas palavras-chave dos utilizadores, não é no entanto válida para o *login*.

2.3 Descrição “validade_ip”

Função que permite validar um IP passado no formato de “pontos e números”.

Para a função um IP válido contém no mínimo 4 números e 3 pontos e no máximo 12 números e 3 pontos. Endereços de rede ou de *broadcast* não são considerados válidos.

2.4 Descrição de “open_socket”

A função *open_socket* permite abrir um *socket* de comunicação e manter o *socket* aberto aceitando e processando clientes que se ligam até que seja negada a condição que mantém o *socket* aberto.

Embora todo o processo de aceitar e tratar pedidos de clientes seja conseguido com um ciclo, apenas um cliente pode ser tratado de cada vez.

2.5 Descrição de "monitor_thread"

Função a executar pela *thread* e que permite a monitorização das sessões presentes no servidor, Securitas, e que permite verificar a validade das sessões e a sua remoção caso tenham gasto o tempo de inactividade disponível.

2.6 Descrição de "proc_pedido"

Função que permite tratar o pedido de um cliente de acordo com o protocolo aplicacional especificado para o servidor *Sercuritas* (ver protocolo aplicacional em "securitas_protocolo.pdf").
A função começa por estipular um tempo máximo para que o cliente envie dados através do socket aberto utilizando a função "alarm()". Após ter lido com sucesso a mensagem enviada, a função verifica se o pedido está de acordo com o protocolo. Caso a mensagem seja válida o pedido é processado e os resultados, a existirem, são enviados para o cliente. Se a mensagem for considerada inválida, a ligação ao cliente é fechada sem que a este seja indicada a razão.
Se o cliente não enviar dados dentro do prazo estipulado a ligação é terminada.

3 Justificação de Opções Tomadas

Na implementação do servidor foram considerados os seguintes pressupostos:

- Nas entradas do ficheiro de contas são permitidos todos os caracteres excepto o "ponto-e-virgula", ";" para o *login*, e o "cardinal", "#". Estes caracteres são considerados parte do formato do ficheiro e são tratados de forma especial. O "ponto-e-vírgula" é no entanto válido para a palavra-chave;
- Nas entradas do ficheiro de máquinas, tal como nas do ficheiro de contas, não são permitidos *logins* de utilizador com "ponto-e-vírgula", nem o uso de "cardinal" juntamente com a informação fundamental ao servidor;
- Não são permitidas entradas diferentes para máquinas iguais no ficheiro de máquinas, isto é, sempre que no ficheiro de contas existirem duas entradas com o mesmo IP, apenas a última é guardada;
- Quando for necessário resolver um nome de uma máquina, apenas o IP principal é assumido como válido. Desta forma pretende-se minimizar falhas de segurança que poderiam advir de ser dada permissão a todos os IP pelos quais o nome resolvido respondeu. Embora esta solução não seja a correcta e se torne demasiado redutora foi considerada a que seria possível implementar.

Para as estruturas de dados do servidor foram tidas em conta características como a velocidade no acesso, a relevância dessa velocidade para a execução do servidor, a necessidade de manter dados ordenados, a forma como essas estruturas são acedidas durante a execução do *securitas*.
Tendo em conta as características especificadas em cima, resolveu-se utilizar para guardar os dados do ficheiro de máquinas uma tabela de *hash* cuja chave é o endereço IP da máquina, o uso da tabela de *hash* prende-se com o facto de ser necessário um acesso rápido aos dados e uma tabela de *hash* é a estrutura que nos permite um acesso rápido e acima de tudo constante aos dados que contém. A tabela de *hash* das máquinas contém uma tabela de I com os logins dos utilizadores que possuem acesso à máquina e uma lista com as pastas disponíveis para pesquisa nessa máquina. Foi escolhida uma tabela de *hash* para guardar os logins pelas mesmas razões invocadas na escolha da tabela de *hash* de máquinas, nesta tabela a chave e o valor são os mesmos pois apenas interessa saber se determinado login se encontra não tabela ou não. A escolha da lista para as pastas advém do facto de não ser necessário pesquisar essa mesma lista, apenas percorrer para obter as pastas, logo uma lista seria a estrutura mais indicada.

Para os dados das sessões mantidas no servidor optou-se, novamente, pelo uso de uma tabela de *hash*, pelos mesmos motivos de velocidade de acesso constante independente do número de elementos guardados.

4 Bibliografia

R.Stevens; UNIX Network Programming Prentice-Hall; 2ª Edição 1998.

Apontamentos teóricos da disciplina de Sistemas Distribuídos e Paralelos.

Fichas práticas de disciplina de Sistemas Operativos.

Os *sites*:

- http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html
- http://publib16.boulder.ibm.com/pseries/en_US/aixprggd/genprogc/signal_mgmt.htm