

# Sistemas Operativos 2015/2016

## Projeto

### Servidor DNS

#### 1. Objetivos do trabalho

- Desenvolver um servidor de DNS (*Domain Name System*) simplificado, usando a linguagem de programação C.
- Explorar os mecanismos de gestão de processos, *threads*, comunicação e sincronização entre processos no sistema operativo Linux.

#### 2. Visão geral do funcionamento do servidor

A Figura 1 apresenta uma visão geral do funcionamento e das interações do servidor a desenvolver.

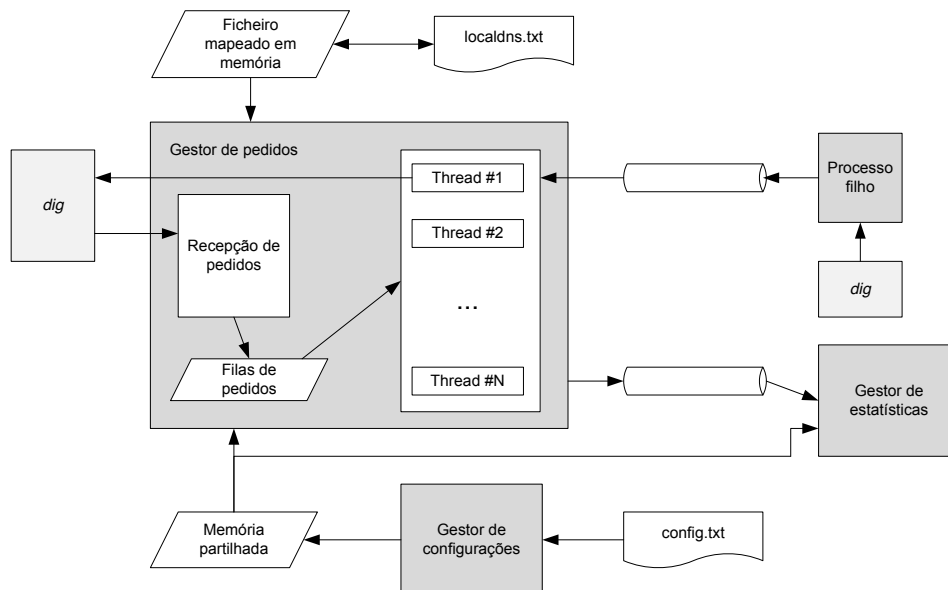


Figura 1 - Visão geral do funcionamento do Servidor

Para realizar pedidos ao servidor DNS será usado como processo cliente o comando `dig` do Linux. Este comando envia pedidos de resolução de nomes ao processo gestor de pedidos e recebe como resposta um endereço IP (*Internet Protocol*).

O servidor de DNS a implementar é constituído por 3 processos principais e 1 auxiliar, que são responsáveis pelas seguintes funcionalidades:

- O gestor de pedidos recebe e trata todos os pedidos de DNS . É responsável pela aceitação de novos pedidos, pela gestão das suas prioridades, pelo escalonamento das *threads*, pela execução de processos externos para tratamento dos pedidos de nomes externos ao domínio local e por escrever informação de *log* para o *named pipe* das estatísticas (no arranque do servidor e a cada pedido recebido e tratado).
- O processo gestor de configurações é responsável pela leitura dos parâmetros de configuração do servidor a partir de um ficheiro de configuração. A informação de configuração determina o funcionamento do servidor e ficará acessível ao processo gestor de pedidos e ao processo gestor de estatísticas.
- O processo gestor de estatísticas é responsável pela leitura das estatísticas geradas pelo processo gestor de pedidos e pela sua escrita no ecrã.
- O “processo filho” é um processo criado e usado de cada vez que é necessário executar o comando externo `dig` para tratar os endereços externos ao domínio local.

As funcionalidades a implementar encontram-se descritas com maior detalhe na secção seguinte.

### **3. Funcionalidades a implementar**

#### **3.1. Envio de pedidos**

Para enviar pedidos será usado um utilitário de DNS já existente em Linux: `dig` (*domain information groper*). Este comando recebe pedidos de resolução de nomes, faz esses mesmos pedidos a servidores de DNS e devolve em resposta os endereços IP que correspondem a esses nomes.

Neste trabalho, o pedido de resolução de nomes efetuado pelo `dig` será recebido pelo processo gestor de pedidos, que funcionará como um servidor de DNS. Uma versão reduzida deste servidor é fornecida com este enunciado e explicada na Secção 4.

Por omissão o `dig` usa o porto 53 para comunicar com os servidores DNS. No caso deste trabalho o servidor irá usar o porto que for fornecido como parâmetro.

Para saber mais detalhes sobre o comando `dig` consulte o manual *online*.

### 3.2. Receção de pedidos

O processo gestor de pedidos é responsável por receber os pedidos de resolução dos nomes em endereços IP. Após a recepção de um pedido as seguintes operações são efetuadas:

1. Cada um dos pedidos é interpretado de modo a saber se é um pedido de um endereço do domínio local (presente no ficheiro de configuração). Pedidos pertencentes ao domínio local têm prioridade superior a todos os outros.
2. Todos os dados do pedido, incluindo a sua origem (`struct sockaddr_in dest` no código fornecido), são inseridos numa de **2 filas** de acordo com a sua prioridade. Existe uma fila para processos de maior prioridade e outra para os “normais”.

### 3.3. Tratamento de pedidos

Os pedidos existentes nas filas são executados por *threads* que correm dentro do processo gestor de pedidos, **privilegiando primeiro aqueles com maior prioridade e em segundo a sua antiguidade**. O número de *threads* a criar é definido na configuração do servidor. Cada uma das *threads* é responsável por validar o pedido, i.e., verificar se a resolução desse domínio é permitida (o domínio local é sempre permitido). De seguida existem 2 opções:

- Se o pedido for válido e corresponder ao domínio local, a *thread* acede a um ficheiro de texto (`localdns.txt`) que foi inicialmente mapeado em memória (*memory mapped file*) e que contém os nomes e respectivos endereços IP de todo o domínio local.
- Se o pedido for válido e corresponder a um domínio externo, a *thread* executa o comando `dig` existente no Linux, o qual permite a resolução de endereços usando servidores de DNS externos. Para isso, a *thread* cria um processo auxiliar e recebe a resposta através de um *pipe*. O resultado desse pedido é depois tratado de modo a extrair o endereço IP. Caso o nome corresponda a vários endereços IP escolhe-se um deles para apresentar ao utilizador. Caso o endereço IP não exista será devolvido o IP 0.0.0.0 ao processo cliente.

A resposta, que consiste no endereço IP que corresponde ao nome recebido, ou na mensagem de erro, é enviada ao processo cliente que realizou o pedido.

### 3.4. Parâmetros de configuração

A configuração do servidor usa um ficheiro de texto (`config.txt`) e uma zona de memória partilhada. O ficheiro de texto contém todas as informações necessárias ao bom funcionamento do servidor e pode ser editado em qualquer altura. O seu carregamento para memória partilhada é realizado automaticamente durante o arranque do servidor. O processo gestor de pedidos (bem como as várias *threads*) poderá consultar a informação de configuração sempre que necessário através dessa zona de memória partilhada.

Caso se verifique a necessidade de alterar as configurações durante o funcionamento do servidor têm de ser efetuadas as seguintes operações:

1. Envio de um sinal SIGUSR1 ao processo de gestão de configurações. Ao receber esse sinal o processo altera uma *flag* em memória partilhada de modo a informar que o servidor se encontra em modo de manutenção e notifica o processo gestor de pedidos que a *flag* foi alterada. Durante o período de manutenção o processo gestor de pedidos apenas responde a pedidos de resolução do domínio local.
2. O utilizador edita o ficheiro de texto com as configurações.
3. O utilizador envia novamente um sinal SIGUSR1 ao processo de gestão de configurações indicando que a nova configuração está pronta, podendo ser lida.
4. O processo de gestão de configurações lê as novas configurações para memória partilhada e sai do modo de manutenção notificando o processo gestor de pedidos.
5. O processo gestor de pedidos aplica as modificações decorrentes das alterações à configuração. Com o servidor em execução, as únicas alterações aplicáveis são respeitantes à identificação do domínio local e da lista de domínios autorizados.

O ficheiro de configuração deverá permitir definir os seguintes parâmetros de configuração do Servidor:

- Número de *threads* da *pool*
- Lista de domínios autorizados
- Domínio local
- *Named pipe* para onde enviar as estatísticas

Exemplo de um ficheiro de configuração (`config.txt`):

```
Threads = 5
Domains = uc.pt; edu
LocalDomain = so.local
NamedPipeEstatisticas = statistics
```

Existirá também um ficheiro (`localdns.txt`) que guarda os IPs e nomes locais. Este ficheiro é fornecido com o enunciado podendo ser-lhe acrescentadas novas linhas.

Exemplo de algumas linhas do ficheiro de consulta do DNS:

```
mywebserver.so.local    192.168.1.20
fileserver.so.local     192.168.1.10
```

### **3.5. Estatísticas de funcionamento**

As estatísticas de funcionamento do servidor são enviadas pelo gestor de pedidos para o *named pipe* das estatísticas. Essa informação é de seguida lida pelo processo de gestão de estatísticas que se encarrega de as processar e agregar, escrevendo a cada 30s no ecrã a seguinte informação:

- Hora de arranque do servidor;

- Número total de pedidos;
- Número de pedidos recusados;
- Número de endereços do domínio local resolvidos;
- Número de endereços de domínios externos resolvidos;
- Data e hora da última informação obtida.

### **3.6. Arranque e terminação do Servidor**

Ao arrancar, o processo gestor de pedidos cria todas as estruturas necessárias ao bom funcionamento do servidor:

1. Cria o processo gestor de configurações e o processo gestor de estatísticas;
2. Lê o ficheiro de configurações para memória partilhada;
3. Cria as *threads* de acordo com o que está nas configurações;
4. Mapeia em memória o ficheiro de nomes/endereços IP que contém todos os dados sobre o domínio local;
5. Cria todos os recursos de comunicação e sincronização considerados necessários;
6. Envia para o *named pipe* das estatísticas a hora de arranque.

O servidor deverá estar igualmente preparado para terminar, após a recepção, por parte do processo gestor de pedidos, de um sinal do tipo SIGINT. Nessa altura todos os processos devem ser terminados e deverá fazer-se a limpeza no sistema de todos os recursos partilhados.

## **4. Uso do código fornecido**

O código fornecido com este enunciado permite interpretar pedidos de tradução de endereços (especificamente A-Records) a servidores de DNS e a sua devolução.

### **Como testar:**

1. Compilar o código do servidor DNS fornecido
2. Arrancar o servidor com o comando `./dnsserver 53000`
3. Executar o comando `dig @127.0.0.1 -p53000 www.teste.com`
  - a. Este comando tenta obter o endereço IP que corresponde ao domínio `www.teste.com` através da execução do comando `dig`, usando o servidor DNS localizado no endereço 127.0.0.1 (máquina local) no porto 53000 (porto que está a ser usado por omissão no servidor fornecido).
4. Deverá receber como resposta o endereço 10.0.0.2 (endereço que o código fornecido devolve por omissão e que terá de ser depois alterado).

## 5. Checklist

Processo	Tarefa	Quantidade de trabalho estimada
Gestor pedidos	Utilizar o <code>fork()</code> para criar o Gestor de configurações e o Gestor de estatísticas	5%
	Criar <i>pipe</i> para estatísticas	2%
	Mapear o ficheiro de nomes e IPs em memória	4%
	Criar e mapear a região de memória partilhada	4%
	Ler corretamente os pedidos	5%
	Escrever corretamente a resposta aos pedidos	5%
	Criar, inicializar e gerir a <i>pool</i> de <i>threads</i>	10%
	Sincronização de <i>threads</i> e dos acessos a recursos partilhados	15%
	Gerir filas de pedidos	6%
	Validar pedidos	5%
	Processar pedido local	5%
	Processar pedido externo	5%
	Prevenir interrupções indesejada por sinais e fornecer a resposta adequada ao SIGINT	5%
	Funcionamento de acordo com as configurações existentes em memória partilhada	5%
Gestor de configurações	Ler para memória partilhada o ficheiro de configuração	5%
	Implementar modo de manutenção	5%
Gestor de estatísticas	Ler estatísticas do <i>pipe</i> e escrevê-las no ecrã	5%
Geral	Deteção e tratamento de erros.	2%
	Terminação dos processos filhos quando o processo gestor de pedidos termina. Libertação de recursos e limpeza ao terminar a aplicação.	2%

### Notas importantes: ESQUEMÁTICO E MAKEFILE

- **Não será tolerado plágio ou qualquer outro tipo de fraude.** Tentativas neste sentido resultarão na **classificação de ZERO VALORES** no projeto.
- Em vez de começar a programar de imediato pense com tempo no problema e estruture devidamente a sua solução.
- Inclua na sua solução o código necessário à deteção e correção de erros.
- Evite esperas ativas no código e assegure a terminação limpa do servidor, ou seja com todos os recursos utilizados a serem removidos.
- Utilize um *makefile* para simplificar o processo de compilação.
- Inclua informação de *debug* que facilite o acompanhamento da execução do programa, utilizando por exemplo a seguinte abordagem:

```
#define DEBUG //remove this line to remove debug messages
(...)
#ifdef DEBUG
printf("Creating shared memory\n");
#endif
```

## 6. Metas, entregas e datas

Data	Meta	
<b>Semana de 19/10/2015</b>	Demonstração intermédia	Os alunos deverão apresentar o seu trabalho nas aulas PL, preparando uma demonstração do trabalho efetuado até ao momento, que deverá representar no mínimo 25% do trabalho global necessário para terminar o projeto.
<b>29/11/2015</b>	Entrega final	<p>O projeto final deverá ser submetido no InforEstudante, tendo em conta o seguinte:</p> <ul style="list-style-type: none"> <li>• Os nomes e números dos alunos do grupo devem ser colocados no início dos ficheiros com o código fonte. Inclua neste local também informação sobre o tempo total despendido (pelo dois elementos do grupo) no projeto.</li> <li>• Com o código deve ser entregue um relatório sucinto (no máximo 2 páginas A4) que explique as opções tomadas na construção da solução.</li> <li>• Crie um arquivo no formato ZIP com todos os ficheiros do trabalho.</li> </ul>
<b>Semanas de 30/11/2015 a 18/12/2015</b>	Defesa	Defesas funcionais em grupo, nas aulas PL.