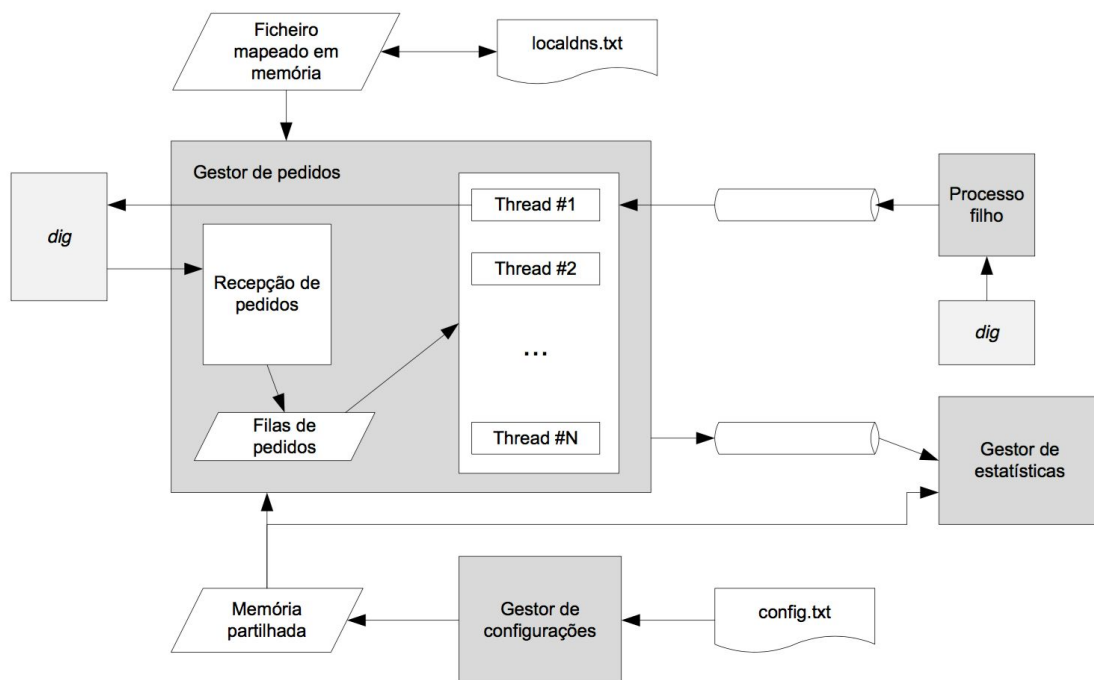




**FCTUC** FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

# Servidor DNS

Sistemas Operativos  
2015/2016



Rita Maria Faria Almeida | ritaa@student.dei.uc.pt | 2012169259

Tomás Morgado de Carvalho Conceição | tmdcc@student.dei.uc.pt | 2012138578

# 1. Manual do Programador

A pasta principal do projeto contém dois ficheiros de texto: o ficheiro *config.txt* que contém os parâmetros de configuração do servidor e o ficheiro *localdns.txt* que contém os nomes e respetivos endereços IP de todo o domínio local. Nessa mesma pasta encontram-se o *makefile* e o ficheiro de código *dnsserver.c*.

## 1.1. Ficheiro dnsserver.c

Como ilustrado na figura, o servidor DNS está dividido em três processos: o processo principal, o processo gestor de configurações e o processo gestor de estatísticas. O processo principal cria os dois outros processos, sendo então o pai e os outros os processos filho.

Definem-se estruturas para a memória partilhada das configurações, uma estrutura para as estatísticas, para os pedidos e para a pool de threads.

Para controlar o acesso à memória partilhada das configurações e à memória mapeada dos domínios locais do ficheiro *localdns.txt*, recorre-se a dois semáforos.

A função “*gestao\_config*” guarda em memória partilhada as informações acerca do servidor, número de threads da pool, lista de domínios autorizados, domínio local e named pipe para onde enviar as estatísticas, lidas do ficheiro *config.txt*. Antes de ser lida qualquer informação do ficheiro *config.txt*, é mapeado o ficheiro *localdns.txt* para memória (memory mapped file).

A função “*gestao\_estatisticas*” é a função que se encarrega de, a cada 30 segundos, escrever no ecrã a informação enviada pelo gestor de pedidos para o named pipe das estatísticas. Esta informação é enviada pelo gestor de pedidos (main) e de seguida é lida pelo processo de gestão de estatísticas, como a hora de arranque do servidor, número total de pedidos, pedidos recusados, endereços do domínio local resolvidos, endereços de domínios externos resolvidos e a data e hora da última informação obtida.

A função “*cria\_pool*” inicializa a pool de threads, criando as threads necessárias para a gestão dos pedidos, inicializando as flags necessárias e o semáforo mutex.

A função “*func\_thread\_pool*” é utilizada por cada uma das threads da pool para verificar as filas de espera de pedidos e responder de forma adequada conforme haja ou não pedidos. Este acesso das threads às filas é controlado por um semáforo *mutex*

que apenas permite o acesso de uma thread de cada vez. Caso as filas de espera estejam vazias, a thread aguarda pela variável de condição *not\_empty*. Assim que haja pedidos numa das filas de espera (dando prioridade à fila de pedidos locais), o pedido na cabeça da fila é atendido e, caso a fila fique vazia, a variável de condição *empty* é sinalizada.

À medida que são recebidos pedidos, é verificado na *main* se o pedido é local ou externo. Caso não seja nenhum destes o pedido é recusado. A função *adiciona\_pedido* assegura que os pedidos são adicionados à respectiva fila de espera (na verificação na *main* caso o pedido seja local a variável *tipo\_dominio* é igualada a 1, caso seja externo a variável é igualada a 2) pela ordem pela qual devem ser atendidos. O pedido a ser atendido é sempre o que se encontra na cabeça da fila de espera.

A função “*ctrl\_c*” é responsável por, após a recepção do sinal do tipo *SIGINT* por parte do processo principal, destruir a pool de threads (aguardando pela conclusão dos pedidos a ser atendidos naquele instante e pelo fim de execução de todas as threads da pool), terminar todos os processos e libertar todos os recursos.

Como a thread das estatísticas é uma thread fora do processo principal, foi criada a função “*ctrl\_c\_esta*” para a destruir.

A função “*modo\_manutencao*” tem como função a alteração da flag em memória partilhada após a recepção do sinal *SIGUSR1*, que notifica o gestor de pedidos (main) que a flag foi alterada e que durante o período de manutenção, ou seja até que seja recebido outro sinal *SIGUSR1*, apenas são respondidos pedidos de resolução do domínio local, sendo que os restantes pedidos são recusados. Enquanto o modo de manutenção está ativado é possível alterar os ficheiro das configurações. Após a manutenção estar concluída é enviado pelo utilizador um sinal *SIGUSR1* notificando que a nova configuração está pronta, é lido novamente o ficheiro *config.txt* e sai do modo de manutenção notificando o processo principal.

A função “*send\_local*” recebe o pedido de um domínio local, vai procurar o IP ao memory mapped file e caso o encontre envia como resposta. Caso não o encontre envia “0.0.0.0” como resposta.

A função “*send\_extern*” recebe o pedido de um domínio externo e executa o “*dig*” do pedido. Caso o “*dig*” devolva o IP desse domínio externo é o que se retorna, senão é retornado 0.0.0.0.

## 2. Manual do Utilizador

Visto que existe um *makefile*, basta aceder à pasta do projeto e executar o comando *make* na linha de comandos para compila e correr o programa.

Para executar um pedido de resolução de um domínio, é necessário chamar a função *dig* com o endereço local e o porto (por exemplo 53000), seguido do domínio a resolver:

```
dig @127.0.0.1 -p 53000 ww5.so.local
```