

## Relatório do Projeto Final de Sistemas Operativos – Simulador de Offloading

Eduardo Figueiredo 2020212310

Fábio Santos – 2020212310

### Ficheiro de configuração

Este ficheiro contém todos os dados necessários para o arranque e funcionamento do simulador. Este contém o tamanho da lista de Tarefas a serem executadas pelos Edge Servers, o tempo máximo para que o nível de performance dos servidores seja elevado, o número de servidores e as características dos mesmos. Foram adicionadas proteções no código tais que devem existir 2 ou mais Edge Servers e que as características destes estejam escritas de forma correta.

### Mobile Node

Processo que gera as tarefas a serem enviadas para os servidores executarem. Estas tarefas são enviadas por um *named pipe* de nome “TASK\_PIPE”. Optamos por mandar o ID da tarefa, o número de instruções e o tempo máximo de execução no formato de string pois iria facilitar o processo de leitura quando fosse necessário distinguir uma tarefa de um comando enviado para o “TASK\_PIPE” (“EXIT” e “STATS”). Se ao correr o executável o número de parâmetros estiver incorreto o Mobile Node não inicia apresentando um erro.

### System Manager

Este processo é encarregue de ler o ficheiro de configurações e começar o programa de acordo com ele. É também ele que cria os processos auxiliares Task Manager, Monitor e Maintenance Manager, o named pipe, a Message Queue para as manutenções, a memória partilhada e todos os mecanismos de sincronização. Uma vez que o System Manager é o processo principal é ele que fica encarregue de capturar os Sinais. Como existem vários processos a correr em simultâneo havia o risco de ler o sinal mais que uma vez. Para isso não acontecer foram ignorados todos os sinais antes da criação dos processos auxiliares e apenas foram redirecionados para os respetivos handlers depois do código de cada um dos processos.

### Task Manager

É Task Manager que lê do named pipe. Se as strings lidas forem tarefas gere-as e distribui-as por cada um dos Edge Servers através de um (unnamed) pipe. Se for um comando válido é chamada a mesma função que é usada quando se redireciona um sinal.

Este processo cria um processo para cada Edge Server e 2 threads: uma thread scheduler e uma thread dispatcher.

### Scheduler

Esta thread analisa as tarefas e dá-lhes prioridades. A prioridade de uma tarefa é o tempo máximo de execução, isto é, quando menos tempo a tarefa demora a ser executada, mais prioritária é. A tarefa que chega ao scheduler e é inserida na lista de tarefas logo na posição correta. A cada inserção é feita uma reavaliação de forma a atualizar o tempo máximo de execução tendo em conta o tempo que já passou desde a sua inserção na lista. Se a lista estiver cheia, a tarefa é descartada.

É usado um semáforo sempre que é inserida ou retirada uma tarefa da lista (*sem\_fila*), de forma a não inserir ou retirar a mesma tarefa mais que uma vez. São também usadas duas variáveis de condição: *cond\_monitor* para informar o Monitor que foi recebida ou retirada uma tarefa para este saber se deve acionar ou desligar o modo High Performance; *cond\_dispatcher* para ativar o dispatcher sempre que uma tarefa entra na fila.

### Dispatcher

Thread que verifica os Edge Servers disponíveis e distribui as tarefas por eles. Em modo Normal as tarefas são enviadas para o vCPU com menor capacidade de forma a poupar energia. Se o Edge Server disponível não tiver capacidade para executar a tarefa, esta é enviada para outro Edge Server. Se apenas existir um servidor livre e este não for capaz de executar a tarefa, a tarefa é descartada. Quando o sistema está em modo High Performance o mesmo processo é feito para ambos os vCPU's dos servidores.

### Maintenance Manager

Por vezes os servidores têm de entrar em manutenção. Para isso o Maintenance Manager verifica quantos servidores estão em manutenção, uma vez que não podem ficar todos em manutenção ao mesmo tempo. Após essa verificação, é escolhido um servidor para entrar em manutenção. Através da Message Queue é enviada a informação para o servidor que este vai entrar em manutenção. Para evitar que a lista seja lida por ordem de chegada e para ter a certeza de que o servidor certo lê a mensagem que lhe é enviada, o parâmetro que distingue qual Edge Server vai ler da Message Queue é o índice do Edge Server +1, de forma a evitar o valor 0.

### Edge Server

Processo que contém uma thread para cada vCPU e que executa as tarefas recebidas pelo pipe que lhe corresponde. A(s) thread(s) executa(m) uma rotina sempre que uma tarefa chega ao servidor, isto é, chega uma tarefa, é criada a thread do vCPU correspondente, é executada a tarefa e a thread acaba, chegando outra tarefa é feito o mesmo procedimento. Sempre que uma tarefa é terminada, o Dispatcher e o respetivo Edge Server são informados que o vCPU está livre através de variáveis de condição (*cond\_dispatcher* e *cond\_manutencao*).

O servidor nunca entra em manutenção a meio de uma tarefa. Para isso, antes da execução de uma tarefa o Edge Server verifica se tem alguma mensagem na Message Queue que lhe corresponde. Se não tiver prossegue para a execução de uma nova tarefa. Isto acontece devido à flag *IPC\_NOWAIT* que torna a função de leitura não bloqueante. Uma vez entrada na manutenção, são alterados os valores das variáveis que indicam que o servidor e os respetivos vCPU's estão ativos e que estão em manutenção. É esperado o tempo da manutenção e estas variáveis voltam ao normal informando o Maintenance Manager terminou a manutenção através de variáveis de condição.

### Monitor

Processo responsável por alternar o sistema entre o modo Normal e o High Performance. Sempre que o tamanho da fila ultrapasse os 80% e o tempo da tarefa ser executada ultrapasse o *MAX\_WAIT* é acionado o modo High Performance e sempre que baixe para os 20% volta ao modo Normal.

Para alcançar esse resultado são utilizados semáforos e variáveis de condição (*cond\_monitor*).