Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Sistemas Operativos

Ano Letivo de 2023/2024

Orquestrador de Tarefas

Flávia Alexandra Silva Araújo (A96587)
Miguel Torres Carvalho (A95485)

4 de maio de 2024

SO

Resumo

» Atualizar o resumo do relatório de acordo com as mudanças na segunda parte do trabalho.

Índice

1 Arquitetura do Serviço **1**
- 1.1 Diagrama de Arquitetura do Serviço . . . 1
- 1.2 Descrição dos Módulos Desenvolvidos . . . 1
  - 1.2.1 Orquestrador (*orchestrator.c*) . . . 1
  - 1.2.2 Cliente (*client.c*) . . . 1
  - 1.2.3 Pedido (*request.c*) . . . 1
  - 1.2.4 Comando (*command.c*) . . . 1
  - 1.2.5 Número de Tarefa (*task_nr.c*) . . . 1
- 1.3 Diagramas de Comunicação entre Servidor e Cliente . . . 1
  - 1.3.1 Execução e Agendamento de Tarefas . . . 1
  - 1.3.2 Estado de Tarefas . . . 3

2 Argumentos da Interface de Linha de Comandos **4**
- 2.1 *Orchestrator* . . . 4
- 2.2 *Client* . . . 4

3 Avaliação de Políticas de Escalonamento **5**
- 3.1 **FCFS** - *First Come First Served* . . . 5
- 3.2 **SJF** - *Shortest Job First* . . . 5
- 3.3 **PES** - *Priority Escalonation Scheduling* . . . 5

4 Testes Desenvolvidos **7**

5 Conclusões **8**

Anexos **9**
- [I] Documentação do Código Desenvolvido . . . 9
- [II] Implementação das Políticas de Escalonamento . . . 9

Índice de Figuras

1.1 Diagrama de Comunicação para Execução e Agendamento de Tarefas 2

1 Arquitetura do Serviço

1.1 Diagrama de Arquitetura do Serviço

1.2 Descrição dos Módulos Desenvolvidos

1.2.1 Orquestrador (*orchestrator.c*)

1.2.2 Cliente (*client.c*)

1.2.3 Pedido (*request.c*)

1.2.4 Comando (*command.c*)

1.2.5 Número de Tarefa (*task_nr.c*)

1.3 Diagramas de Comunicação entre Servidor e Cliente

1.3.1 Execução e Agendamento de Tarefas

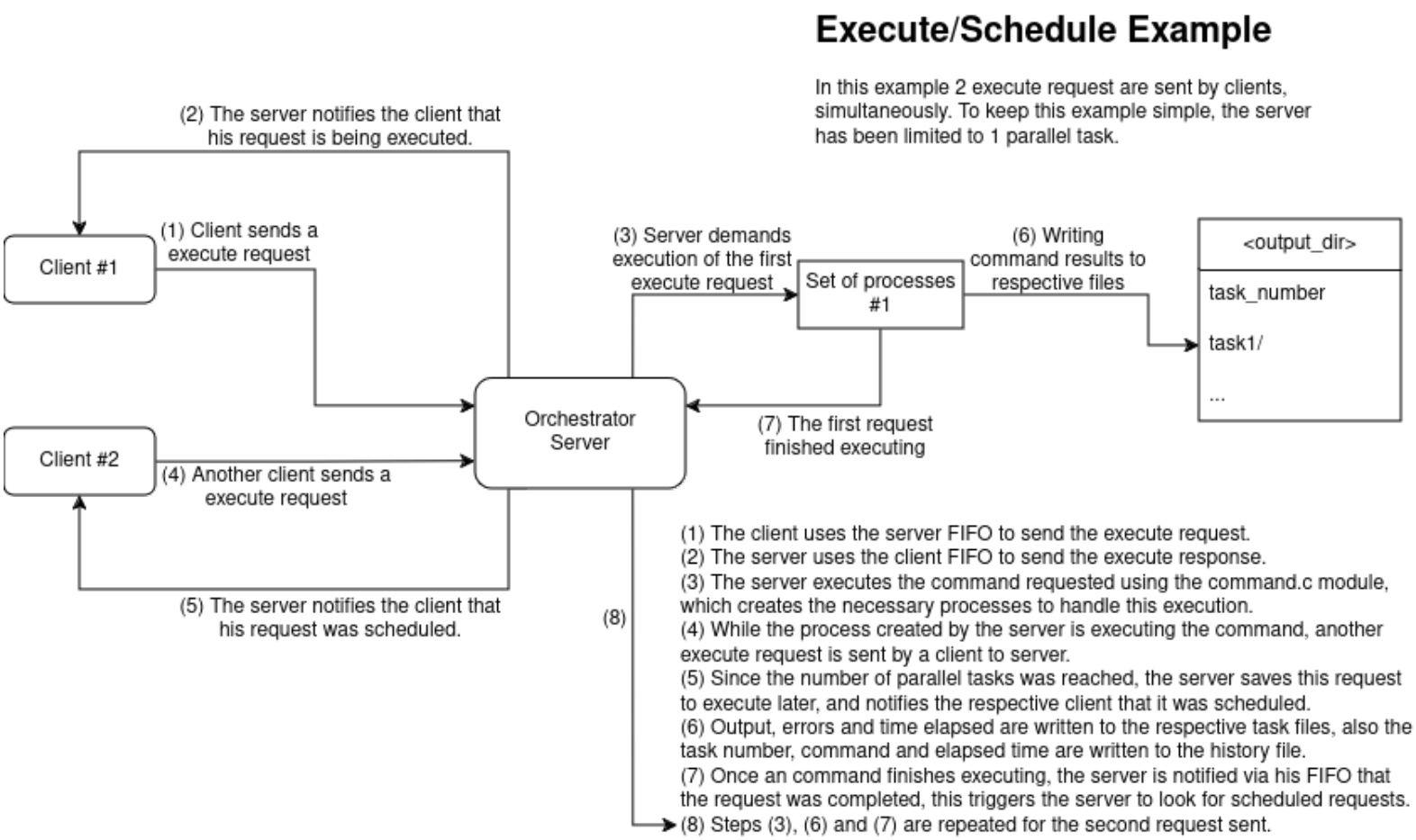

Figura 1.1: Diagrama de Comunicação para Execução e Agendamento de Tarefas

1.3.2 Estado de Tarefas

2 Argumentos da Interface de Linha de Comandos

2.1 *Orchestrator*

2.2 *Client*

3 Avaliação de Políticas de Escalonamento

No trabalho desenvolvido, foram implementadas três políticas de escalonamento de tarefas. Estas serão aprofundadas nas secções seguintes, assim como as suas avaliações práticas. Por favor consulte o anexo [II] Implementação das Políticas de Escalonamento para mais detalhes de como for realizada a implementação destas políticas.

» Atualizar a introdução da avaliação de políticas de escalonamento.

3.1 FCFS - *First Come First Served*

Nesta política, as tarefas são executadas por ordem de chegada, o que não é eficiente em termos de tempo de espera e execução. No entanto, é uma política simples e fácil de implementar.

» Escrever sobre a avaliação prática desta política + Imagem

3.2 SJF - *Shortest Job First*

A política *Shortest Job First* é uma política de escalonamento não preemptiva que seleciona a tarefa com o menor tempo de execução, este tempo é passado como argumento na opção *execute* do cliente e repesenta uma estimativa do tempo que a tarefa demorará a ser executada.

Esta política é eficiente em termos de tempo de espera e execução, mas pode levar a situações de *starvation*, que ocorrem quando tarefas com tempos de execução maiores são sempre adiadas, e num caso extremo, quando surgem constantemente tarefas com tempos de execução menores, estas de maior duração nunca serão executadas.

Existem várias maneiras de lidar com situações de *starvation*, como por exemplo a criação de uma especificação de tempo máximo de espera para cada tarefa, ou a definicação de um numero máximo de tarefas que podem ser executadas antes de uma tarefa com tempo de execução maior. Outra solução seria a implementação de uma política de escalonamento preemptiva, que permitiria a interrupção de tarefas em execução, para dar lugar a tarefas com tempos de execução menores.

» Escrever sobre a avaliação prática desta política + Imagem

3.3 PES - *Priority Escalonation Scheduling*

De forma a evitar situações de *starvation*, foi implementada a política *Priority Escalonation Scheduling*. Nesta, as tarefas são executadas de acordo com a sua prioridade, que é definida pelo cliente na opção *execute*, simliarmente como o tempo estimado que é passado como argumento na política *SJF*.

As tarefas com maior prioridade são executadas primeiro, evitando situações de *starvation*. As prioridades ficam a critério do cliente, possibilitando uma maior versatilidade na execução de tarefas, permitindo ao cliente definir a prioridade adequada para cada tarefa que deseja executar. No entanto, esta flexibilidade não garante a eficiência da execução das tarefas, caso estas não sejam corretamente priorizadas.

» Escrever sobre a avaliação prática desta política + Imagem

4 Testes Desenvolvidos

5 Conclusões

TODO escrever sobre interrupções do servidor e como este poderia lidar com estas

Após o desenvolvimento deste serviço, foram cumpridos todos os objetivos propostos no enunciado do trabalho, desde a implementação de uma interface de linha de comandos tanto para o servidor, como para o cliente, que permite a execução de tarefas do utilizador de forma assíncrona com a possibilidade de encadeamento de programas com *pipes* e a consulta do estado das tarefas, assim como a implementação de um ficheiro *log* para guardar as tarefas executadas e a implementação de um sistema com várias políticas de escalonamento e a avaliação prática destas. Também foi desenvolvido um conjunto de testes que permitem verificar o correto funcionamento do serviço.

Após a afirmação da possibilidade da terminação ou interrupção do servidor pelo professor regente foi despertada a curiosidade de como o servidor poderia lidar com estas situações.

Num estudo apronfundado desta unidade curricular, uma solução seria a implementação de um sistema de interrupções no servidor, que permitiria a este lidar com situações de terminação ou interrupção de forma mais eficiente, utlizando os sinais do sistema operativo para lidar com estas situações, evitando a perda de tarefas em execução e agendadas.

Para tal, seria necessário lidar com tarefas em execução, guardando o estado destas em memória, e permitindo a sua correta terminação ou interrupção, assim como para as tarefas em espera. Estas poderiam ser guardadas numa fila de espera, que seria consultada sempre que uma tarefa terminasse, permitindo a execução da próxima tarefa na fila.

Em suma, o desenvolvimento deste serviço permitiu a aplicação prática dos conhecimentos adquiridos ao longo do semestre, assim como a exploração de novas funcionalidades e conceitos, que permitiram a implementação de um serviço robusto e versátil, que permite a execução de tarefas de forma assíncrona, com a possibilidade de encadeamento de programas e a consulta do estado das tarefas.

Anexos

Clique nos links abaixo para aceder aos anexos.

[I] Documentação do Código Desenvolvido

[II] Implementação das Políticas de Escalonamento