
Projeto (Fase II)

1 Introdução ao Projeto

A leitura prévia do enunciado da fase anterior é fundamental e deve ser considerada como introdução a este enunciado. O objetivo geral desta fase é:

- a utilização de semáforos para controlar o acesso aos buffers, em particular, aos 3 buffers de comunicação entre cliente/intermediário/empresas e capacidade portuária.
- a interação com o sistema de ficheiros e a utilização das funções de tempo. Particularmente pretende-se ler o conteúdo do ficheiro de entrada e gerar os ficheiros de saída e de log. Pretende-se também que sejam utilizados os mecanismos de medição de tempo e ativação de alarmes.
- a alteração do cliente para poder efetuar N operações e para guardar e imprimir estatísticas dos tempos de execução das mesmas.

2 Estruturas de dados

Nesta fase do projeto são introduzidas as estruturas de dados `prodcons`, `file` e `op_time`.

A estrutura `prodcons` dá suporte ao controlo de acesso a cada um dos 3 buffers. Para isso oferece 3 semáforos para cada buffer (total de 9), nomeadamente: `full`, `empty` e `mutex`. O semáforo `full` é ativado (post) pelo produtor e testado (wait) pelo consumidor. O semáforo `empty` é ativado (post) pelo consumidor e testado (wait) pelo produtor. Por exemplo, um cliente é visto como um produtor quando insere (produz) uma descrição de operação no buffer `BDescricao`, enquanto um intermediário será um consumidor quando acede a este buffer para ler uma descrição de operação. O semáforo `mutex` assegura a exclusão mútua no acesso ao buffer. Adicionalmente, esta estrutura armazena um outro semáforo do tipo `mutex` para garantir exclusão mútua no acesso à capacidade portuária.

A estrutura `file` regista os nomes dos ficheiros de entrada, resultados e log, e os respetivos *handles*. Os nomes dos ficheiros são recebidos através da consola pelo `main.c` e passados para a função `file_begin`. Os respetivos *handles* são ponteiros para estruturas de dados que são manipuladas por funções da biblioteca. Após a abertura de um ficheiro, é obtido um *handle* que é um parâmetro a passar a cada função que manipule esse ficheiro. A estrutura dos ficheiros de entrada e resultado foi explicada no enunciado da Fase I. Mas, a estrutura do ficheiro de log é explicada na Seção 4.

A estrutura `op_time` (definida no `main.h`) guarda o tempo de execução `time`, de uma operação executada pelo cliente `cliente`. A variável `tempos` da estrutura `statistics` passa a guardar um array de `op_time` em memória partilhada, permitindo assim guardar os tempos de execução de todas as operações e qual o cliente que as invocou. Estes tempos podem, no fim da execução do `SO_AdmPor`, ser percorridos para imprimir estatísticas como mínimos, máximos e médias de tempos de execução de cada cliente.

3 Depuração com o utilitário logPlayer

Quando o SO_AdmPor é executado com um determinado ficheiro de entrada, para além do ficheiro de resultado, e também gerado um ficheiro de log. O comando logPlayer permite ver o conteúdo (ou parte) deste último ficheiro de log gerado pelo sistema. O ficheiro de log pode ser muito útil para fazer depuração do programa por mostrar o conteúdo dos 3 buffers (i.e., descrição, pedido e agendamento) sempre que são alterados. Para gerar este ficheiro no SO_AdmPor deve ser utilizada a opção "-l" seguida do nome do ficheiro de log a criar.

O ficheiro de log é constituído por registos separados por uma sequência de 4 inteiros com todos os bits a um. Cada registo contém o tempo decorrido (double), a etapa em que foi gerado (int), o id de quem o gerou (int) (cliente, intermediário ou empresa) e o conteúdo das posições ocupadas dos 3 buffers, respetivamente: BDescricao, BPedido e BAgendamento (múltiplos de 4 int).

A etapa é um número natural cujo valor corresponde, respetivamente às seguintes etapas no acesso aos buffers:

1. Cliente elabora descrição de operação;
2. Intermediário lê descrição de operação;
3. Intermediário escreve pedido de operação;
4. Empresa lê pedido de operação;
5. Empresa escreve proposta de agendamento;
6. Cliente lê proposta de agendamento.

O conteúdo de cada buffer está separado por uma sequência de 2 inteiros com todos os bits a 1 e mais 2 inteiros com todos os bits a 0. O conteúdo de cada buffer é constituído por 0, ou mais, entradas de 4 int cada. Cada entrada contém 4 identificadores que são: id da operação (int), id do cliente (int), id do intermediário (int) e id da empresa (int). O logplayer pode ser chamado passando apenas o nome do ficheiro de log a mostrar. Podem também ser especificados, na linha de comando, os seguintes filtros:

- etapa a visualizar
- id do cliente/intermediário/empresa a visualizar
- o número do primeiro registo a mostrar
- o número do último registo a mostrar.

Considera-se que os registos a visualizar são numerados sequencialmente a partir de 1, em dependência dos filtros escolhidos. Para conhecer as opções disponíveis basta executar o logPlayer sem parâmetros. Os ficheiros de log deverão ser colocados no diretório SO_AdmPor/tests/log.

4 Objetivos

Esta fase envolve a escrita de código nos ficheiros prodcons.c, file.c, time.c e cliente.c nas zonas de desenvolvimento que estão rodeadas com comentários do tipo "// =====".

Para tal, os **alunos terão de fazer download e usar os novos Ficheiros do Projeto 2 disponibilizados nos Moodle**, por terem sido alterados alguns ficheiros *.h e o so.o.

Neste projeto devem ter o especial cuidado de tratar o resultado de qualquer chamada de sistema. Os alunos devem reaproveitar os ficheiros makefile e script.sh desenvolvidos na Fase I do projeto e usar os ficheiros memory.c e main.c fornecidos na Fase II do projeto.

4.1 Utilização de semáforos (prodcons.c)

De uma forma geral devem ser utilizadas as funções POSIX adequadas para operações sobre semáforos e devem ser evitadas situações de espera ativa e interbloqueio entre os processos.

4.1.1 Modelo produtor-consumidor

Os objetivos específicos desta seção seguem os do Modelo produtor-consumidor e são os seguintes:

1. Criação e inicialização de todos os semáforos de controlo no acesso aos buffers e capacidade portuária.
2. Controlar adequadamente o acesso aos buffers (circular e de acesso aleatório) e capacidade portuária.
3. Atualizar a capacidade portuária, decrementando-a caso seja superior a zero.
4. Destruição de todos os semáforos criados.

As operações `prodcons_*_begin()` e `prodcons_*_end()` devem utilizar as funções `sem_wait` e `sem_post` de forma a garantir a correta operação do `SO_AdmPor`.

4.2 Interação com o sistema de ficheiros (file.c)

De uma forma geral devem ser utilizadas as funções da biblioteca C standard I/O adequadas para operações sobre ficheiros, medição de tempo e alarmes.

4.2.1 Leitura e Escrita de Ficheiros

Os objetivos específicos desta seção referem-se à leitura e escrita de ficheiros e são os seguintes:

1. Abrir o ficheiro de entrada em modo de texto e tratar o seu conteúdo. Dado que os campos da estrutura `Config` do tipo `char *` são preenchidos automaticamente pela função `ini_parse_file`, inicializar, a partir destes, os restantes campos da mesma estrutura.
2. Abrir o ficheiro de saída (especificando-o como parâmetro de entrada do `SO_AdmPor`) em modo de texto.
3. Abrir o ficheiro de log (se especificado como parâmetro de entrada do `SO_AdmPor`) em modo binário. Escrever log para o ficheiro binário.
4. Destruir zonas de memória reservadas dinamicamente.

4.3 Funções de Tempo (time.c)

Os objetivos específicos desta parte são os seguintes:

Permitir armar um alarme, de acordo com o tempo de intervalo passado como parâmetro, para escrita de uma linha de depuração para a consola durante a execução do `SO_AdmPor`.

1. Calcular a diferença entre dois tempos em nanossegundos.
2. Calcular a diferença entre o tempo atual e um tempo inicial em nanossegundos.
3. Registrar o tempo atual.
4. Adormecer os clientes por 1 milissegundo entre pedidos de operações.
5. Adormecer as empresas por uma quantidade de tempo aleatória, entre 1 e 10 milissegundos (simula a execução de um agendamento).
6. Calcular o tempo de execução mínimo, médio e máximo de um dado cliente, através da estrutura `statistics.tempos`.

4.4 Repetição de operações do Cliente (cliente.c)

O ficheiro `cliente.c` deve ser alterado de forma a executar a mesma operação um número de vezes igual a `N` e esperar um milissegundo entre pedidos. Adicionalmente, o tempo de execução de cada operação deve

ser guardado na estrutura `statistics.tempos`. Os alunos podem adaptar o código do ficheiro `cliente.c` da fase anterior, fazendo as alterações descritas em comentário para implementar esta funcionalidade.

5 Teste dos objetivos

Os métodos de teste passam pela análise do ficheiro de entrada (em `tests/in`), dos ficheiros gerados (resultados e log) e em alguns casos também das saídas para a consola (`stdout`). A concretização de cada um dos objetivos desta fase pode ser verificado de acordo com o método indicado nas subsecções seguintes.

5.1 Utilização de semáforos – modelo produtor-consumidor

Os testes para cada objetivo são os seguintes:

1. Verificar que o `SO_AdmPor` continua a funcionar sem qualquer problema.
2. Verificar que o `SO_AdmPor` não bloqueia nem produz resultados errados.
3. Verificar que não se solicitam operações quando não há empresas e que os indicadores da capacidade portuária são coerentes.

5.2 Interação com o sistema de ficheiros – leitura e escrita de ficheiros

Os testes para cada objetivo são os seguintes:

1. Verificar que os vários ficheiros de entrada (disponibilizados como cenários) são lidos e tratados corretamente.
2. Verificar que, para cada ficheiro de entrada, é produzido um ficheiro de resultado correto.
3. Verificar que, para cada ficheiro de entrada, é produzido um ficheiro de log correto. A correção pode ser aferida por comparação com os ficheiros gerados pelo `SO_AdmPor` padrão.
4. Verificar que o `SO_AdmPor` funciona sem gerar erros.

5.3 Funções de Tempo

Os testes para cada objetivo são os seguintes:

1. Verificar que para diferentes intervalos de tempo (passados como parâmetro) são produzidas saídas para a consola em linha com o esperado.
2. Verificar que a evolução dos tempos de execução corresponde ao esperado.
3. Verificar que a empresa espera um tempo que está em linha com os valores indicados.

5.4 Repetição de operações do Cliente

Os testes para cada objetivo são os seguintes:

1. Verificar que os ficheiros gerados (resultados e log) com diferentes valores de `N` estão corretos.
2. Verificar que os indicadores de tempos mínimos, médios e máximos de cada cliente são coerentes.

6 Entrega

Os ficheiros `prodcons.c`, `file.c`, `time.c` e `cliente.c` devem ser entregues até às **23:59** do dia **23/05/2020**, utilizando o link de entrega disponível na página de SO no moodle. Deve-se criar um ficheiro zip com o nome **SO-0XX-proj2.zip**, onde **XX** é o número do grupo, contendo os ficheiros listados acima e um ficheiro `Readme.txt` com as limitações do trabalho. Apenas um dos elementos do grupo deve submeter. Se existirem várias submissões do mesmo grupo, será considerada a mais recente.

Não serão aceites trabalhos por e-mail nem por qualquer outro meio não definido nesta secção. Se não se verificar algum destes requisitos o trabalho é considerado não entregue.