

Trabalho Prático N°2
Sistema de Distribuição Multimédia 'Live'

v1

O processo de distribuição e consumo de informação multimédia do tipo "live" costuma seguir três fases distintas:

1. Geração da informação multimédia numa fonte a um ritmo certo (que pode ir mudando ao longo do tempo, mas não é considerado uma característica dinâmica). O ritmo é definido em blocos de informação por segundo por cada canal, sendo o bloco de informação dependente do tipo, da qualidade e do mecanismo de codificação da informação. Por exemplo, o ritmo de uma fonte multimédia áudio stereo pode ser definido como sendo de codificação PCM de $44100 \times 16 = 705600$ bps por cada canal. Esta fonte disponibiliza a informação a este ritmo para um servidor multimédia que será responsável por retransmitir o sinal para os clientes. Podem existir diversas fontes de informação a alimentar um servidor multimédia. Normalmente, a cada fonte é associado um identificador de circuito/canal. Os clientes podem usar estes identificadores para escolher o circuito/canal de que querem receber a distribuição.
2. Transmissão da informação multimédia do servidor para os clientes, mediante a identificação do circuito/canal. A transmissão pode ser exclusiva no servidor, isto é, cada cliente só pode receber a transmissão dum único circuito/canal, ou pode ser em grupo, isto é, o cliente pode ter um grupo (normalmente limitado) de circuitos/canais que recebe em simultâneo, apesar do utilizador apenas poder reproduzir um deles com qualidade máxima.
3. Reprodução da informação multimédia no módulo reproduzidor do cliente. A reprodução do circuito/canal escolhido pelo utilizador no cliente é feita localmente através duma interface de utilizador. Normalmente, esta interface permite listar os circuitos/canais disponíveis no serviço e também permite ao utilizador escolher um circuito/canal para reprodução, gravação, etc.

O objetivo deste trabalho é o desenvolvimento de um sistema simples, mas completo, com elementos (fonte de informação, servidor multimédia, cliente multimédia e interface de utilizador com reproduzidor multimédia) que implementem os processos básicos das três fases referidas.

Fonte de Informação

O sistema deve incluir um programa que gere informação multimédia a um ritmo certo num determinado formato de codificação e que envie o resultado (ao ritmo definido) para um determinado servidor multimédia (SM). O ritmo e o endereço (IP + Porta UDP) do servidor multimédia devem ser parâmetros

configuráveis (argumentos no arranque, ficheiro de configuração ou menu do programa).

Ou seja, o sistema terá apenas um tipo de fonte de informação, mas pode ter várias fontes a gerar informação ao mesmo tempo, a ritmos diferentes para o mesmo SM.

Sugere-se que a informação a gerar nestas fontes é o valor das amostras duma onda sinusoidal $\sin(x)$ com uma determinada frequência F , amostrada a uma frequência F_a múltipla de F ($F_a = N \cdot F$, $N > 3$). Os valores de F e N são parâmetros de entrada configuráveis e determinam, indiretamente, o ritmo a que a informação é gerada. Por exemplo, se $F = 1\text{Hz}$ e $N = 10$, então $F_a = 10\text{Hz}$. Neste caso, em cada segundo, é gerado um valor (V_i) de amostra a cada centésimo de segundo (ou 100 amostras por segundo) através da fórmula:

$$V_i = \text{int}(1 + (1 + \sin(2 \cdot \pi \cdot i / N)) \cdot 30), \text{ com } 1 \leq i \leq F \cdot N = F_a$$

Assim, cada valor (V_i) deve ser codificado como um número inteiro positivo entre 1 e 61.

Os alunos podem escolher outros tipos de fontes de informação e outras formas de codificação.

Transmissão da fonte de informação para o SM

O protocolo P1 de transmissão duma fonte de informação para o SM deve ser simples, atómico, assimétrico, assíncrono, sem estados e encapsulado em UDP. O protocolo é assimétrico porque não existem interações iniciadas pelo servidor. Assim que programa da fonte começa a arrançar deve começar a sua transmissão ao ritmo definido pelos parâmetros F e N (ou outros que os alunos tenham definido). O SM tem um papel passivo. A fonte não sabe se o SM está a receber corretamente ou não.

Como a quantidade de informação multimédia de cada bloco de informação é pequena (apenas um valor V_i), o ritmo de transmissão de PDU-1 deste protocolo pode ser igual ao ritmo a que a informação é gerada (no exemplo anterior com $F_a = 10\text{Hz}$, teremos um ritmo de 10 PDU/segundo). Cada PDU-1 encapsulado num datagrama UDP deve conter os respetivos valores de i e de V_i , além dos valores F e N (que serão sempre iguais). Como o protocolo não tem estados, é assíncrono e não confirmado, para o servidor saber com quantos valores V_i contar em cada segundo e poder saber se recebeu os valores V_i na ordem certa, o PDU deve conter também a identificação P do período da função $\sin(x)$, isto é, cada período da função (em que são gerados N valores V_i) é identificado por um número P que se inicia a 0 (o valor zero é apenas usado na primeira vez) e vai sendo incrementado até um valor máximo igual a M , sendo então reiniciado a 1. Por razões de segurança (uma vez que o protocolo é não confirmado e assíncrono), durante o primeiro período ($P=0$), logo após o arranque do programa, todos os valores V_i devem ser iguais a zero (isto transmite uma espécie de aviso ao SM que o programa se iniciou).

Também é útil que a fonte de informação seja associada a um identificador D (string), independente do endereço IP e porta UDP que o programa está a usar. Até possibilita que vários programas contribuam concorrentemente para a mesma fonte, desde que usem o mesmo identificador D .

Assim, em resumo, o PDU-1 deve conter, pelo menos, as seguintes informações: D, i, Vi, P, F, N e M. Além disso, o ficheiro de configuração dos programas que implementam a fonte de informação deve conter as seguintes informações: D, F, N, M e endereço (IP + Porta UDP) do SM para onde enviar a informação.

Os alunos devem definir claramente e com o maior detalhe possível o protocolo P1 (interações possíveis, primitivas e PDU-1), justificando convenientemente as estratégias escolhidas e decisões tomadas.

Servidor Multimédia

O SM é um programa que integra três módulos diferentes:

- Módulo que recebe numa porta fixa toda a informação de todas as fontes. Com a informação recebida das fontes o SM deve gerir uma lista de fontes de informação ativas. Essa lista deve poder ser transmitida aos clientes para que os utilizadores possam escolher a fonte que desejam receber e ver reproduzida. Este módulo não necessita de buffer de entrada. Por cada PDU-1 recebido e processado numa fonte deve ser enviado um outro PDU-2 com a informação correspondente para todos os clientes que tenham requerido (subscrito) a receção dessa fonte (ou canal). Este módulo tem de implementar o protocolo P1 para a comunicação com as fontes de informação.
- Módulo de envio da informação relevante para os clientes que tenham subscrito as fontes. Este módulo é chamado pelo módulo anterior, sempre que esse receba um PDU-1 numa fonte de informação. Este módulo tem de implementar o protocolo P2 para interação com os clientes multimédia. O protocolo P2 é mais complexo que o protocolo P1 pois tem de permitir alguma gestão do fluxo da informação entre o SM e os clientes para além da gestão das subscrições dos canais. Este módulo só lida com o processo de envio da informação multimédia aos clientes através do protocolo P2.
- Módulo de gestão das subscrições. Este módulo precisa de implementar o processo de escuta de comandos dos clientes e o processo correspondente de resposta do protocolo P2. Os clientes devem ter quatro tipos de comandos possíveis à sua disposição: list, info(D), play(D), stop(D). Quando este módulo processa um pedido de list deve responder ao cliente com a lista simples dos identificadores D de todos os canais/fontes disponíveis/ativos. A um pedido de info(D), o SM deve responder com os parâmetros disponíveis sobre a fonte D (D, F, N e M). A um pedido de play(D) o SM deve responder OK se o canal D continuar disponível e além disso, deve incluir o cliente na lista de subscritores do canal D. A um pedido de stop(D) o SM deve responder OK e além disso, deve retirar o cliente na lista de subscritores do canal D (se o cliente não for subscritor ignora o pedido).

Interação entre o SM e os clientes

O protocolo P2 para interação entre o SM e os clientes não é tão simples como o P1 mas, ainda assim, deve ser atómico, assimétrico, assíncrono, sem estados e encapsulado em UDP.

Conforme já foi referido, o protocolo P2 deve implementar dois processos diferentes de interação entre o SM e os clientes utilizando o PDU-2:

- Transmissão de dados dum canal D para um cliente - quando o SM recebe um PDU-1 com informação multimédia relevante, deve pegar nessa informação e formar um novo PDU-2 e enviar, o mais rapidamente possível, uma cópia para todos os clientes subscritores desse canal D. Os PDU-2 também devem ser encapsulados em datagramas UDP. Para além da informação contida nos PDU-1, os PDU-2 devem acrescentar, no mínimo, um campo de identificação do tipo de PDU-2 (por exemplo, tipo data). Neste processo de transferência não existe qualquer tipo de confirmação ou garantia de ordenação.
- Gestão de subscrições de canais - o SM pode receber vários tipos de comandos (ou tipos de PDU-2) dos clientes (list, info, play e stop) de maneira a que estes possam iniciar e terminar subscrições dum canal/fonte, conforme referido anteriormente. Para cada um destes tipos de comandos o SM deve enviar uma resposta específica (response-list, response-info, response-play, response-stop). Associado a cada comando deve existir um identificador de comando (e que deve ser incluído em todos os PDU-2 que não sejam do tipo data). Este número é gerado no cliente para cada PDU-2 de comando enviado ao SM. O SM deve usar no PDU-2 de resposta este identificador para que o cliente saiba identificar o comando correspondente à resposta recebida (isto é necessário porque o protocolo é assíncrono e não confirmado).

Os alunos devem definir claramente e com o maior detalhe possível o protocolo P2 (interações possíveis, primitivas e PDU-2), justificando convenientemente as estratégias escolhidas e decisões tomadas.

Clientes Multimédia

Os clientes multimédia são responsáveis por três módulos lógicos, mais ou menos distintos:

Um módulo para receber os PDU-2 do tipo data, retirar a informação relevante e introduzi-la num buffer de entrada que é utilizado pelo módulo de reprodução da informação na interface do utilizador (UI).

Um módulo de processamento da informação no buffer de entrada de modo a permitir uma experiência consistente da informação multimédia no UI. Os utilizadores devem experienciar a visualização dos valores V_i ao ritmo a que foram gerados na fonte correspondente, com o menor atraso possível. A forma mais simples de implementação é imprimir os valores V_i como linhas de V_i caracteres cada, ao ritmo de F_a linhas/segundo (uma espécie de desenho duma onda sinusoidal com "scroll" vertical na linha de comandos).

Um módulo que permita ao utilizador, no UI, ver a lista de canais disponíveis, escolher um para reprodução ou para parar a reprodução.

Otimizações

Podem criar-se fontes de informação mais interessantes que uma simples onda sinusoidal.

O protocolo P2 pode incluir um mecanismo de segurança para o caso dum cliente ser subscritor dum canal e o programa cliente terminar a sua execução (o utilizador pode simplesmente aborrecer-se e sair do programa ou terminar abruptamente a sua execução). Nesse caso, deixa de ser necessário sobrecarregar o SM com a transmissão desse canal para aquele cliente. Uma forma de poder detetar esse fenómeno é o SM acrescentar no PDU-2 do tipo data um campo de créditos atribuídos a um cliente, que é máximo quando ele faz a subscrição num canal ativo. Por cada PDU-2 de data enviado a esse cliente o número dos créditos que tem atribuídos é decrementado. Se o número de créditos do cliente chegar a zero o cliente é retirado da lista de subscritores do canal e a transmissão desse canal para esse cliente termina. Por outro lado, o cliente, à medida que vai recebendo os PDU-2 do tipo data vai vigiando o número de créditos disponíveis e se quiser continuar a receber a transmissão envia outro pedido de subscrição antes do número de créditos chegar a zero.

Relatório e outras recomendações

O trabalho deve ser desenvolvido em grupos de, no máximo, dois alunos.

Sugere-se que o código utilize apenas funções normalizadas da linguagem C, de preferência da norma 217 ISO (STD 17), e ou funções desenvolvidas pelo grupo de trabalho. Não use APIs, funções ou excertos de código de terceiros que não tenham sido fornecidos ou aprovados pelo docente.

Opcionalmente os alunos podem utilizar outras linguagens de programação que não o C, como Java ou Python. Os alunos que usarem a linguagem C, C++ ou Rust para desenvolver o trabalho receberão uma bonificação adicional.

O código deve ser claro e usar convenções de nomeação de variáveis, tipos, funções e constantes. O código deve ser estruturado numa forma o mais modular possível, sem complexidades desnecessárias, e permitir reutilização sempre que possível.

A qualidade e correção do código não se mede pelo seu tamanho.

Os alunos devem documentar/explicar o código criado através de comentários nas secções mais relevantes e no relatório. Todos os ficheiros do código devem ter um cabeçalho com a informação relevante que identifique os seus autores e explique as principais funções criadas, tipos de dados e variáveis usadas, etc.

O relatório deve incluir:

- Na primeira página, o título do trabalho e a identificação dos autores (incluindo fotografia), universidade, curso, unidade curricular e data de entrega;
- Um índice do conteúdo;
- Uma secção com a discussão das estratégias escolhidas, as opções tomadas e os mecanismos adotados, incluindo eventuais otimizações;
- Uma secção com a explicação e análise crítica (apenas) das principais funções implementadas, os seus principais méritos e as suas limitações mais importantes;
- Uma secção com a análise crítica dos resultados dos testes efetuados, tentando explicar o mais detalhadamente as razões lógicas para os

resultados encontrados;

- Uma secção de conclusões que inclua uma eventual discussão sobre o que gostava de ter feito melhor ou de ter acrescentado e não conseguiu;
- Uma lista de eventuais referências bibliográficas, artigos científicos ou recursos informais na web e que tenham sido úteis.

O relatório é para ser avaliado pelo docente por isso não se deve incluir informação genérica e irrelevante que o docente já conhece. Deve aspirar-se concisão e clareza. Sempre que se incluir uma afirmação importante no contexto do relatório e que seja de autoria de terceiros, ou que seja baseada diretamente em afirmações de terceiros ou concluída de informação retirada de recursos alheios, deve referenciar-se corretamente essas autorias ou proveniências e acrescenta-las na lista das referências.

O relatório pode incluir algumas partes relevantes do código quando estas ajudam às análises e justificações apresentadas. Não se deve integrar código desnecessário no texto do relatório. Sempre que possível, comente-se antes o próprio código.

Todo o material entregue deve juntar-se num único ficheiro zip. Este ficheiro zip deve conter um ficheiro PDF com o relatório e todos os ficheiros do código do projeto. O nome do ficheiro zip deve ser igual a SRAM-2022-2023-TP2-Número_Aluno_A-Número_Aluno_B.zip, como, por exemplo, SRAM-2022-2023-TP2-83974-87766.zip.

Por fim, é recomendável que, durante a defesa do trabalho, se responda honesta e concisamente apenas às questões colocadas por forma a que as sessões de apresentação não se arrastem para além dos 20-30 minutos.