

Exemplos de Execução Trabalho Prático 2024.2

Este documento apresenta alguns exemplos de execuções do sistema. A fim de facilitar a explicação, as tabelas a seguir detalham o passo a passo dos comandos de entrada (**em negrito**) e as informações que devem ser impressas em tela em cada instante de tempo.

Abertura de comunicação entre Servidores(Peer-2-Peer)

A tabela 1 apresenta um cenário de comunicação entre três servidores. Como o escopo do trabalho limita o peering para apenas 2 servidores, tal controle pode ser observado no fluxo de comunicação.

Tempo	Terminal 1	Terminal 2	Terminal 3
t ₁	./server 40000 50000		
t ₂	No peer found, starting to listen...		
t ₃		./server 40000 60000	
t ₄	Peer 5 connected		
t ₅		New Peer ID: 5	
t ₆		Peer 9 connected	
t ₈	New Peer ID: 9		
t ₉			./server 40000 70000
t ₁₀			Peer limit exceeded
t ₁₁			(Como estamos executando todos os servidores em uma mesma máquina, um erro de binding ocorrerá, pois a porta 40000 já está em uso. Neste caso, o programa deste servidor pode ser encerrado).

Tabela 1 - Cenário exemplo de abertura de comunicação Peer-2-Peer e controle de número máximo de conexões

Abertura de comunicação de cliente com servidor

A tabela 2 apresenta um cenário de comunicação entre dois servidores e dois clientes. Os dois servidores realizam peering entre si, e cada cliente se conecta a ambos os servidores.

Tempo	Terminal 1 (Peer ID: 9)	Terminal 2 (Peer ID: 5)	Terminal 3 (Client Loc: 1)	Terminal 4 (Client Loc: 7)
t ₁	./server 40000 50000			
t ₂	No peer found, starting to listen...			
t ₃		./server 40000 60000		
t ₄	Peer 5 connected			
t ₅		New Peer ID: 5		
t ₆		Peer 9 connected		
t ₈	New Peer ID: 9			
t ₉			./client 127.0.0.1 50000 60000 1	
t ₁₀	Client 2 added (Loc 1)			
t ₁₁		Client 14 added (Loc 1)		
t ₁₂			SU New ID: 2 SL New ID: 14	
t ₁₃				./client 127.0.0.1 50000 60000 7
t ₁₄	Client 3 added (Loc 7)			
t ₁₅		Client 15 added (Loc 7)		
t ₁₆				SU New ID: 3 SL New ID: 15

Tabela 2 - Cenário de abertura de comunicação cliente-servidor em um contexto onde ocorre peering entre dois servidores.

Fechamento de comunicação (Peer-2-peer e Cliente-servidor)

A tabela 3 apresenta um cenário de comunicação entre dois servidores e dois clientes. Os servidores e clientes já iniciam-se conectados, e essa tabela pode ser vista como uma continuação da tabela anterior. Será demonstrado o processo de encerramento de comunicação entre peers e entre cliente-servidor. A topologia é a seguinte:

Terminal 1: Servidor SU (9)

Terminal 3: IC Loc 1

Terminal 2: Servidor SL (5)

Terminal 4: IC Loc 7

Tempo	Terminal 1 (Peer ID: 9)	Terminal 2 (Peer ID: 5)	Terminal 3 (Client Loc: 1)	Terminal 4 (Client Loc: 7)
t_1				kill
t_2		Client 15 removed (Loc 7)		
t_3	Client 3 removed (Loc 7)			
t_4				SU Successful disconnect SL Successful disconnect
t_5		kill		
t_6	Peer 5 disconnected			
t_8		Successful disconnect Peer 9 disconnected		
t_9	No peer found, starting to listen...			

Tabela 3 - Cenário de fechamento de comunicação cliente-servidor seguido de fechamento de comunicação peer-2-peer.

Requisição de mensagem entre clientes e servidores

A Tabela 4 apresenta um cenário de comunicação entre dois servidores e dois clientes. Os servidores e clientes já iniciam-se conectados e seguem a topologia apresentada anteriormente. Para a coluna dos servidores, será denotado alguns dos pacotes recebidos (<) e enviados (>) por eles. Fica a critério do aluno imprimir as mensagens de requisição e de resposta no servidor para as funcionalidades implementadas, mas isso deve ser feito na saída de erro do programa.

Tempo	Terminal 1 (Servidor SU)	Terminal 2 (Servidor SL)	Terminal 3 (Cliente Loc 1)	Terminal 4 (Cliente Loc 7)
t ₁			add 2021808080 1	
t ₂	< REQ_USRADD 2021808080 1			
t ₃	> OK 2			
t ₄			New user added: 2021808080	
t ₅			find 2021808080	
t ₆		< REQ_USRLOC		
t ₇		> ERROR 18		
t ₈			User not found	
t ₉				in 2021808080
t ₁₀	< REQ_USRACCESS 2021808080 in			
t ₁₁		< REQ_LOCREG 2021808080 7		
t ₁₂	> RES_LOCREG -1			
t ₁₃				Ok. Last location: -1
t ₁₄			find 2021808080	
t ₁₅		< REQ_USRLOC		
t ₁₆		> RES_USRLOC 7		
t ₁₇			Current location: 7	
t ₁₈				add 2021808080 0

t ₁₉	< REQ_USRADD 2021808080 0			
t ₂₀	> OK 3			
t ₂₁				User updated: 2021808080
t ₂₂				out 2021808080
t ₂₃	< REQ_USRACCESS 2021808080 out			
t ₂₄		< REQ_LOCREG 2021808080 -1		
t ₂₅	> RES_LOCREG 7			
t ₂₆				Ok. Last location: 7
t ₂₇			inspect 2021808080 -1	
t ₂₈		< REQ_LOCLIST 2021808080 -1		
t ₂₉	< REQ_USRAUTH 2021808080			
t ₃₀		< RES_USRAUTH 0		
t ₃₁		> ERROR 19		
t ₃₂			Permission denied	
t ₃₃			add 2020909090 1	
t ₃₄			New user added: 2020909090	
t ₃₅			out 2020909090	
t ₃₆			Ok. Last location: -1	
t ₃₇			inspect 2020909090 -1	
t ₃₈		< REQ_LOCLIST 2020909090 -1		
t ₃₉	< REQ_USRAUTH 2020909090			
t ₄₀		< RES_USRAUTH 1		

t ₄₁		> RES_LOCLIST 2021808080 2020909090		
t ₄₂			List of people at the specified location: 2021808080, 2020909090	

Tabela 4 - Cenário de consultas diversas entre clientes e servidores.