
1 Teste automático remoto de servidores GS

O presente guia contém os procedimentos para execução remota da aplicação ‘player’ na máquina ‘tejo’ para teste dos servidores GS desenvolvidos pelos alunos.

1.1 Introdução

No âmbito do projecto de comunicação usando a interface sockets torna-se necessário testar as aplicações desenvolvidas pelos alunos em comunicação com aplicações que cumpram o protocolo especificado.

Tal é a finalidade do servidor concorrente GS em execução na máquina ‘tejo’, o qual permite testar as aplicações ‘player’ desenvolvidas pelos alunos.

Com a finalidade inversa de testar as aplicações GS desenvolvidas pelos alunos, existe em execução na máquina ‘tejo’ um servidor TCP concorrente (no porto 59000) que invoca localmente a aplicação ‘player’ em resposta a um acesso TCP por **netcat**, originado num computador dos alunos, através do qual se especificam os parâmetros de execução remota da aplicação ‘player’.

Assim é possível executar em simultâneo na máquina ‘tejo’ instâncias da aplicação ‘player’ que enviam mensagens a servidores GS alvo em execução na rede pública ou na rede do Técnico.

Ambas as modalidades de teste acima referidas permitem aos alunos suportar tanto o desenvolvimento das suas aplicações como a componente de autoavaliação.

As aplicações ‘player’ que serão executadas remotamente na máquina ‘tejo’ vão ler as sequências de comandos a executar de *scripts*

predefinidos, escolhidos pelos alunos para testar os seus servidores GS. As instâncias da aplicação ‘player’ invocadas remotamente de forma concorrente para execução no tejo mantêm separação de dados não havendo qualquer interferência entre aplicações ‘player’ em execução simultânea.

A figura que se segue ilustra uma sequência de teste típica.

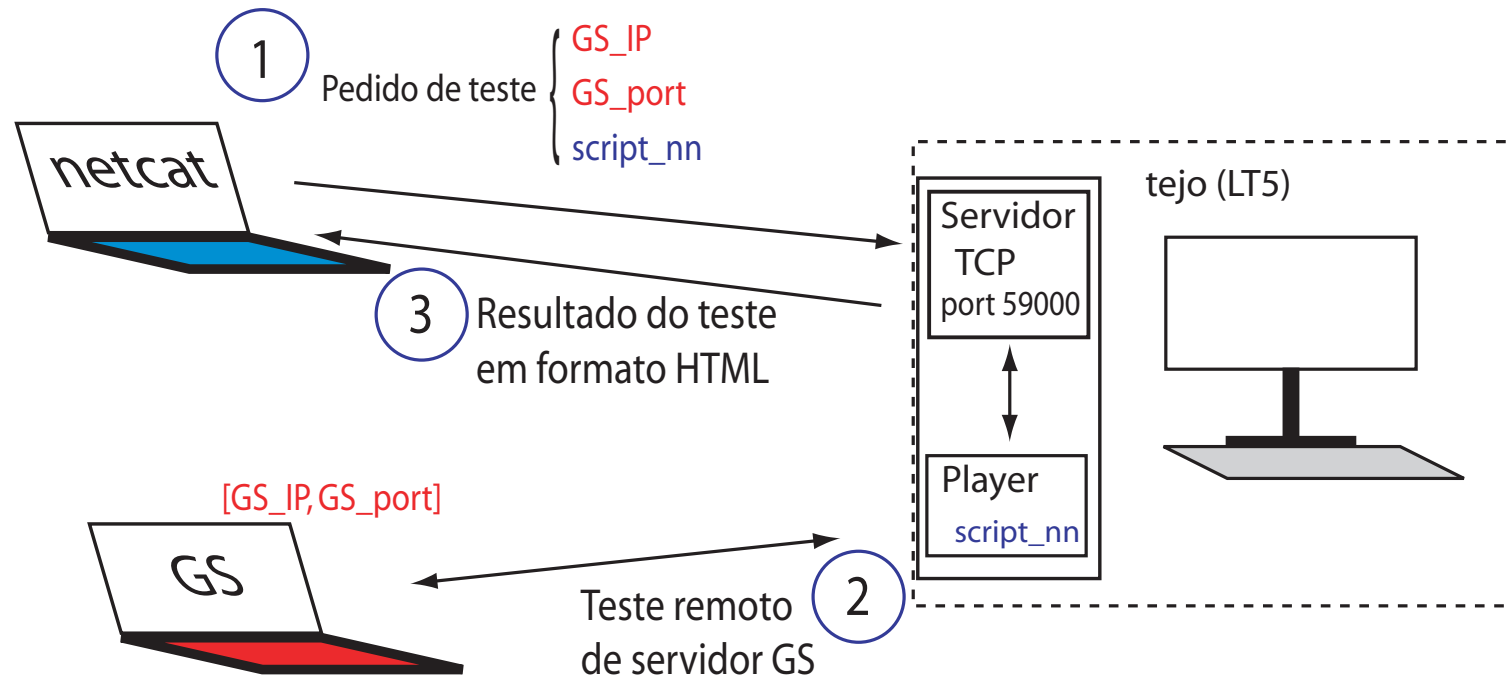


Figure 1: Teste remoto de servidores GS

Para executar um teste subordinado a um dado *script*, os alunos têm de executar a seguinte linha de comando no Linux:

```
echo "target_IP target_port script" | nc tejo.tecnico.ulisboa.pt 59000 > report.html
```

na qual, ‘target_IP’ é o endereço IP da máquina onde executam o seu servidor GS, ‘target_port’ é o porto da sua aplicação GS na referida máquina e ‘script’ é um inteiro com um ou dois dígitos que indica o número do *script* de teste que querem correr (ver *scripts* publicados na

página da disciplina).

A invocação do **nc** não deve ser dissociada da emissão da mensagem de comando, pois entre a aceitação da ligação no servidor e a leitura da mensagem de comando no mesmo existe uma tolerância de tempo muito reduzida.

Na mensagem com o formato apresentado acima, o servidor no tejo espera encontrar o caracter ‘\n’ logo a seguir ao número do *script*, o qual é inserido pelo próprio comando ‘echo’ nas instalações que serviram para elaborar este guia. Caso o comando ‘echo’ não insira o ‘\n’ deve ser usado um procedimento alternativo que o faça.

Os espaços entre campos podem ser em qualquer número desde que o comprimento total da mensagem (incluindo ‘\n’) não ultrapasse os 25 bits. Não são aceites pedidos com os endereços IP público ou privado do tejo, ou igual a 127.*.*.*.

No final da execução, o servidor envia um ficheiro HTML com o resultado da referida execução obtido pela aplicação ‘player’ cuja execução se solicitou. Esse ficheiro pode ser convertido em formato pdf para ser entregue no contexto da auto-avaliação do projecto.

O servidor GS alvo a testar pode ser executado no *sigma* cujo endereço IPv4 pode ser obtido executando o comando *hostname -i*. Note-se que o *sigma* pode ser endereçado por vários IPs. Para testarem o servidor GS no seu domicílio, os alunos devem proceder à configuração de *port forwarding* nos seus *routers*.

Como exemplo, considere-se o servidor GS a testar no porto 58050 do *sigma* com IP=193.136.128.103 para execução do *script* número 6. A linha a executar numa máquina com Linux (podendo ser ela o próprio *sigma*) será:

```
echo "193.136.128.103 58050 6" | nc tejo.tecnico.ulisboa.pt 59000 > report.html
```

Ficando o relatório da execução guardado no ficheiro report.html na máquina na qual se executa o comando **nc**.

1.2 Ambiente de execução

Para os *scripts* de comando da aplicação ‘player’ no tejo invocada remotamente supõe-se que o servidor GS alvo de testes usa as palavras constantes do ficheiro **word_eng.txt** (e apenas essas) bem como todos os ficheiros de sugestão que lhe estão associados os quais se devem localizar na sua directoria de trabalho. Esse ficheiro de palavras e os ficheiros de sugestão respectivos encontram-se comprimidos no ficheiro **Dados_GS.zip** na secção ‘Laboratório’ da página da disciplina.

Na execução dos *scripts*, o servidor GS alvo deve escolher, para a execução de cada *script*, em sequência, as palavras constantes no ficheiro **word_eng.txt**, começando pela primeira.

Um determinado *script* poderá seleccionar a palavra a decifrar emitindo pares de comandos ‘start’/‘quit’, até que o GS esteja posicionado para seleccionar a palavra pretendida.

1.3 Teste de funcionalidades suportadas em UDP

Os scripts numerados de 1 a 11 testam funcionalidades do servidor GS exclusivamente através de comunicações UDP.

Para cada um dos scripts, o GS deve arrancar escolhendo em primeiro lugar a primeira palavra do ficheiro **word_eng.txt** e escolhendo para cada novo pedido de iniciação de jogo as restantes palavras do ficheiro em sequência.

Salientam-se de entre estes 11 scripts aqueles que pela sua extensão justificam uma explicação:

O ‘script_03’ solicita ao GS a escolha de todas as palavras do ficheiro usando uma sequência de comandos ‘start’. O total dos pedidos excede em uma unidade o número de palavras do ficheiro, pelo que a primeira palavra deverá ser de novo seleccionada pelo GS no último jogo. Após cada operação de iniciação de jogo o ‘player’ envia uma jogada ‘guess’ com a palavra respectiva para efeitos de confirmação.

O ‘script_04’ solicita ao GS a escolha da primeira palavra, com o ‘player’ a submeter jogadas ‘play’ certas e jogadas erradas, excedendo estas o número limite de erros. O jogo deve terminar com insucesso.

O ‘script_05’ solicita ao GS a escolha da primeira palavra, com o ‘player’ a submeter jogadas ‘play’ certas e jogadas erradas, não excedendo estas o número limite de erros. O jogo deve terminar com sucesso.

O ‘script_06’ solicita ao GS a escolha da primeira palavra, com o ‘player’ a submeter jogadas ‘play’ certas e erradas e jogadas ‘guess’ erradas, excedendo as erradas o número limite de erros. O jogo deve terminar com insucesso.

O ‘script_07’ solicita ao GS a escolha da primeira palavra, com o ‘player’ a submeter jogadas ‘play’ certas e erradas e jogadas ‘guess’ erradas, não excedendo as erradas o número limite de erros. O jogo deve terminar com sucesso.

O ‘script_08’ solicita ao GS a escolha da quarta palavra, com o ‘player’ a submeter jogadas ‘play’ certas, erradas e repetidas excedendo as erradas o número limite de erros. O jogo deve terminar com insucesso.

O ‘script_09’ solicita ao GS a escolha da quarta palavra, com o ‘player’ a submeter jogadas ‘play’ certas, erradas e repetidas não excedendo as erradas o número limite de erros. O jogo deve terminar com sucesso.

O ‘script_10’ solicita ao GS a escolha da terceira palavra, com o ‘player’ a submeter jogadas ‘play’ certas e erradas alternadamente não excedendo as erradas o número limite de erros. O jogo deve terminar com sucesso.

O ‘script_11’ é uma evolução do ‘script_03’. Apresenta a mesma sequência de jogos, sendo em cada jogo incluídas jogadas ‘play’ certas e erradas antes da emissão do comando guess com a palavra correcta.

1.4 Teste de funcionalidades suportadas em TCP

Nos scripts numerados de 20 a 26 são testadas funcionalidades relacionadas com a comunicação TCP, devendo os alunos ter as comunicações UDP a funcionar adequadamente pois o teste das comunicações em TCP depende do sucesso daquelas.

Para cada um dos *scripts*, o GS deve arrancar escolhendo em primeiro lugar a primeira palavra do ficheiro **word_eng.txt** e escolhendo para cada novo pedido de iniciação de jogo as restantes palavras do ficheiro em sequência.

O ‘script_20’ é uma evolução do ‘script_10’ e testa os resultados da mensagem STA (derivada do comando state) repetidas vezes para um único jogo para a terceira palavra do ficheiro **word_eng.txt**.

O ‘script_21’ é uma evolução do ‘script_11’ incorporando o comando state emitido em cada jogo antes e depois do jogo terminar.

Os *scripts* com os números 22, 23 e 24 testam a resposta do GS alvo ao comando ‘scoreboard’ no ‘player’.

Os registos de *score* devem estar completamente limpos antes da execução destes três *scripts* em sequência.

Os *scripts* com os números 25 e 26 testam a resposta do GS alvo ao comando ‘hint’ no ‘player’.