UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ

INSTITUTO DE MATEMÁTICA – IM
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC

SISTEMA CLIENTE-SERVIDOR
TEMA 1

Disciplina: Sistemas Operacionais
Professores: Thomé e Valéria

Júlio César Machado Bueno	106033507
Luiza Diniz e Castro	107362705
Roberta Santos Lopes	107362886

Sumário

Apresentação

O documento presente tem como principal função estabelecer a documentação requisitada no Trabalho 3 da disciplina. O tema sorteado é conhecido como Sistema Cliente-Servidor. Neste programa são utilizadas técnicas de *Threads* e comunicação por TCP/IP e *sockets.*

Especificações gerais

O Sistema foi concebido de forma a conter dois Programas distintos que possuem o seu funcionamento simultâneo em máquinas distintas. Desta forma temos um Programa Servidor e Programa Cliente. Cada programa possui codificação, uso e propriedades diferentes. Todos são escritos utilizando a linguagem ***Java*** dado a sua fácil implementação de Interface Gráfica além de compatibilidade entre os diversos SO.

Para o funcionamento correto do Sistema devemos ter apenas 1 instância do Programa Servidor sendo executado e até 5 instâncias do Programa Cliente em máquinas distintas da máquina em que o Programa Servidor está sendo executado. O limite de instâncias a serem executadas pelo Programa Cliente foi fixado para que se fosse possível um melhor controle das operações com os arquivos durante a apresentação simulada e também para respeitar os requisitos técnicos do Trabalho. Entretanto, esse limite de instâncias não influencia em nenhum momento a lógica de operação do Sistema e portanto pode ser estendido facilmente.

É necessário também que as máquinas onde as instâncias do Programa Cliente estão sendo executadas, tenham conexão direta a máquina onde está sendo executado o Programa Servidor através do protocolo TCP/IP. Ou seja, o número IP da máquina do Programa Servidor deve ser acessível externamente.

A fim de simplificação, o Programa Servidor oferece aos Clientes somente os arquivos encontrados na raiz do diretório onde o mesmo se encontra em execução. Desta forma os Clientes fazem requisições ao Servidor destes arquivos.

Da mesma forma o Programa Cliente só possui acesso localmente aos arquivos que se encontram em seu diretório raiz de execução.

Regras de Negócio

Dado as especificações técnicas encontradas na descrição do Trabalho e também na lógica a ser implementada no Sistema, temos as seguintes Regras de Negócio para o Programa Servidor e Programa Cliente:

- Servidor
  - O Servidor deve ser a única instância em execução em uma máquina. Caso contrário retornará um erro de recursos indisponíveis.
  - O Servidor sempre será inicializado com a porta 2222 e endereço local (*localhost*, 127.0.0.1).
  - O Servidor não possui interface gráfica.
  - O Servidor só aceita até 5 Clientes conectados a ele. Caso um novo Cliente tente a conexão ele retornará um erro.
  - O Servidor terá como saída somente os comandos enviados pelos Clientes e o registro de novas conexões ou Clientes desconectados.
  - Ao receber o comando CTRL+C o Servidor é encerrado.
  - Um Recurso (arquivo) só pode sofrer um comando destrutivo caso nenhum outro Cliente esta executando um comando sobre o Recurso específico.

- Cliente
  - O Cliente deve ser a única instância em execução em uma máquina.
  - Ao iniciar o Cliente, somente ao campo *host* e ao botão “Conectar” estarão habilitados ao Usuário.
  - O Usuário deve preencher o campo *host* com um endereço de IP para um Servidor válido. Caso contrário retorna um alerta de falha na conexão.
  - Ao preencher o campo *host* com um endereço de IP válido e clicar no botão “Conectar” uma conexão remota deve ser criada.
  - Ao criar uma conexão remota, o campo *host* e o botão “Conectar” devem estar desabilitados.
  - Ao criar uma conexão remota, exceto o campo *host* e o botão “Conectar” devem estar habilitados.
  - O Usuário só executa uma funcionalidade representada por cada botão de cada vez.
  - O Usuário só executa uma funcionalidade em um único arquivo selecionado.
  - Ao terminar a execução do Programa Cliente sem efetuar previamente o comando representado pelo botão “Desconectar” é enviado uma mensagem de erro ao Servidor.
  - Se um Recurso estiver bloqueado por uso pelo Servidor qualquer comando sobre o Recurso não terá efeito.

Utilizamos então, durante a codificação, estas Regras assim como práticas de Engenharia de Software. Estas práticas incluem a notificação de ações do Sistema para o Usuário de forma clara, assim como a segurança da manipulação de dados baseada nas ferramentas disponibilizadas na Linguagem. Desta forma garantimos a integridade do Sistema com o mínimo de intrusão. Ou seja, utilizamos a linguagem convenientemente para garantir o funcionamento correto do Sistema. Isto será de grande importância para garantir que os Clientes possam agir de forma concorrente ao Servidor.

Modelagem

A Modelagem realizada do Sistema foi concebida baseando-se em nos conceitos adquiridos em sala de aula, nas Regras de Negócio apresentadas além da realização do Trabalho 2 e também no material bibliográfico.

Podemos caracterizar as Entidades do Sistema da seguinte forma:

- **Servidor**
  É a Entidade responsável pela criação do *socket* de comunicação e uso da porta do Sistema assim como a criação de *Threads* Atendentes responsáveis pelo controle de execução e ações do Servidor. Também é responsável pelo controle de acesso a recursos. Desta forma, a função básica do Servidor é gerenciar o número e execução das *Threads* Atendentes.

- **Atendente**
  É uma *Thread* inicializada pelo Servidor e é responsável por 1 conexão direta a um Cliente específico e todo o processamento requerido pelo mesmo. Ou seja, é de fato o que realiza as operações no servidor. Também é responsável pelo registro de ações realizadas por cada Cliente.

- **Cliente**
  É uma aplicação dissociada do Servidor que possui sua execução em uma máquina distinta e é responsável por tanto a conexão e envio de informações ao Servidor quanto o processamento de informações locais.

- **Recurso**
  É o item a ser disponibilizado pelo Servidor para os Clientes ou pelos Clientes para o Servidor. Neste Sistema, todo Recurso é um arquivo encontrado na raiz do diretório de execução de cada Programa. No caso do Recurso do Servidor, este possui a característica de ser bloqueante durante ações destrutivas ou de transferência.

A seguir temos a Figura 1 que ilustra o funcionamento do Sistema e mostra suas principais entidades:

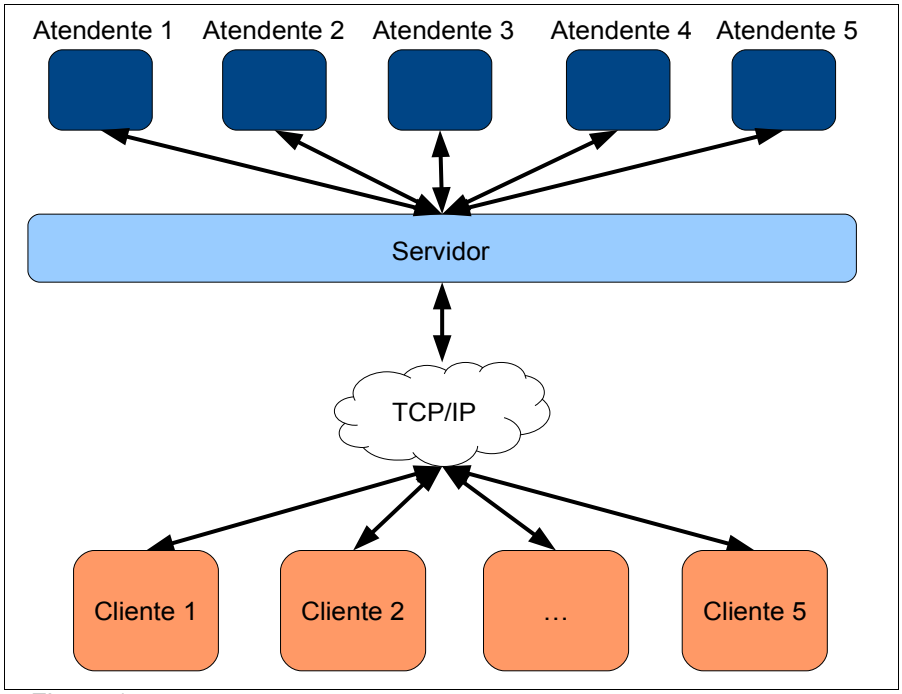

Figura 1

Podemos então apresentar um fluxograma que exemplifica o fluxo básico da execução do Programa Servidor e a conexão de um Programa Cliente através de um Atendente.

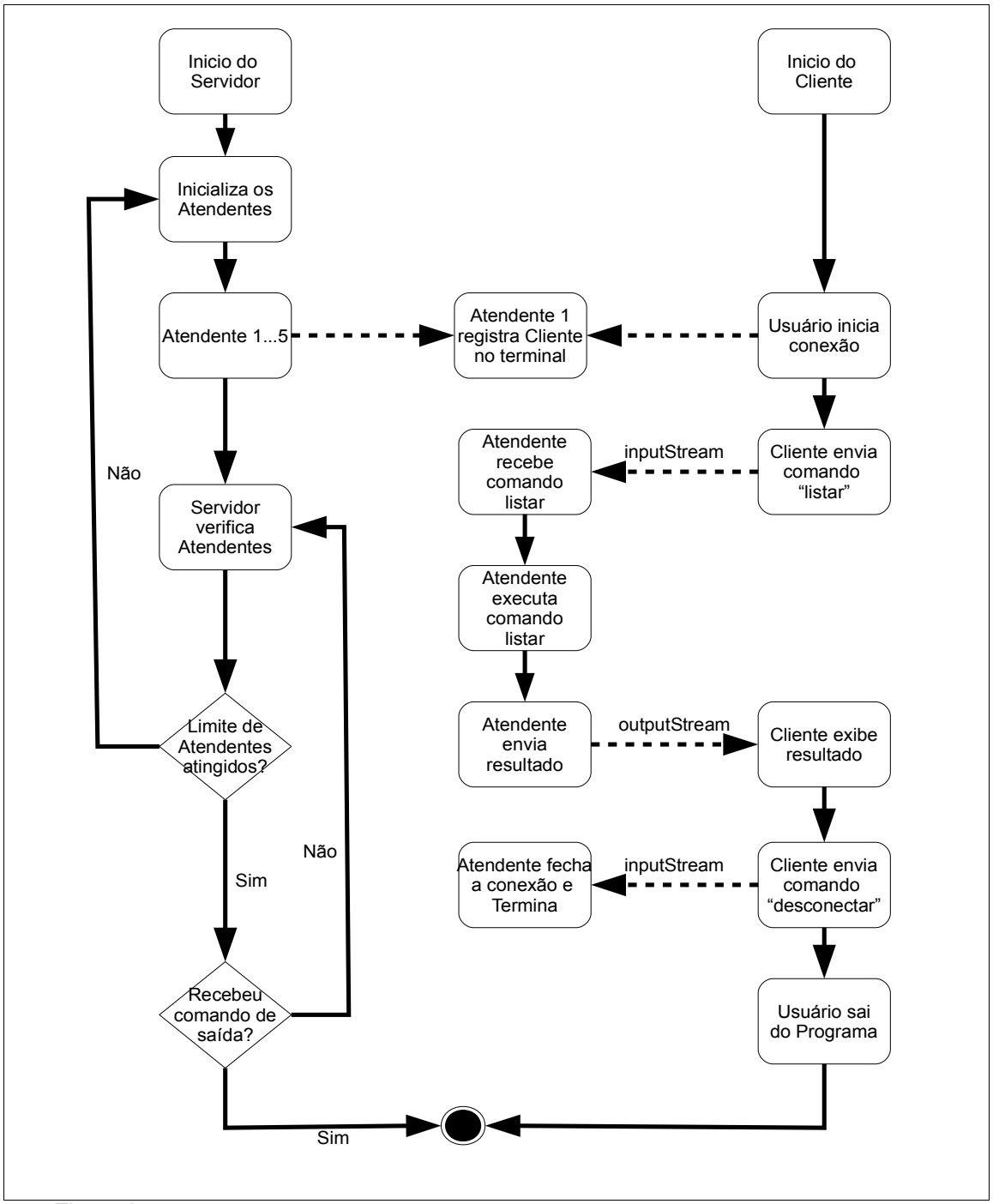

Figura 2

Lógica de Semáforos

Como os Programas Clientes atuam de forma concorrente no Servidor através dos Atendentes, é necessário a implementação de uma Lógica que permita que os Recursos seja utilizados da melhor forma possível. Neste caso, nos referimos a Recursos como os arquivos localizados no mesmo diretório do Servidor.

Como um exemplo de problema de concorrência, podemos ter um Recurso sendo transferidos ao Cliente conectado ao Atendente 1, enquanto o Cliente conectado ao Atendente 2 pode requerer que este mesmo Recurso (arquivo em transferência) seja removido do Servidor. Neste caso, dados as Regras de Negócio implementadas, é necessário que a transferência seja concluída para que remoção ocorra.

Podemos identificar então um evento de acesso a uma área de Exclusão Mútua. Desta forma o Sistema conta com a implementação de uma Lógica de Semáforos incluída no Servidor a fim de garantir a integridade do Sistema. Esta lógica é implementada da seguinte forma:

Interface e Operação

A interface de cada Programa é diferencia. Para o Programa Servidor, foi especificado que este operasse somente em modo texto enquanto o Programa Cliente possui uma interface gráfica GUI multi-plataforma. Desta forma, a operação do Servidor se dá através do console somente. Cada ação do Servidor representativa para o Sistema é expressa como saída do Terminal.

A seguir temos o exemplo de execução apresentado no Fluxograma da Modelagem (Figura 2):

```
Servidor criado em localhost
Foram criadas 5 Threads Atendentes para 5 Clientes
Um novo Cliente se conectou a Thread Atendente 1
Foi executado o comando <ls -p>
Cliente conectado ao Atendente 1 desconectou-se do Servidor
```

Neste exemplo podemos observar que:

- O Servidor indica a sua criação. (Linha 1)
- O número de *Threads* Atendentes em execução. (Linha 2)
- A conexão de um Cliente a *Thread* Atendente. (Linha 3)
- A requisição e execução de uma funcionalidade. (Linha 4)
- A desconexão de um Cliente. (Linha 5)

Esta interface exibe o status do processamento de cada um dos Produtores e Consumidores. Além disso, durante a execução do programa, é exibido um registro em forma de *log* que informa todas as ações do simulador, assim como dos recursos a cada produção ou consumo dos mesmos. Dessa forma:

- Quando o produtor produz um recurso, é inserida um registro indicando:
  - Número de identificação do Recurso produzido. Afim de simplificação, cada novo Recurso tem o seu identificador igual a sua posição de produção.
  - Total de recursos já produzidos até o momento.

        - Total de recursos disponíveis para consumo.

    - Quando o Consumidor consome um Recurso, é inserido um novo registro indicando:
        - O número de identificação do Recurso consumido.
        - Total de recursos ainda disponíveis para consumo.

Também é exibido um painel informativo contendo valores dos Semáforos utilizados para controlar o acesso a Região Crítica do sistema. Nesse caso, temos um Semáforo utilizado pelos Consumidores e outro pelos Produtores. Os valores podem ser 0 ou 1 para os Produtores e, dependendo da versão selecionada, 0 ou 1 ou, valores inteiros entre 0 e 5 para o Consumidores.

É possível selecionar entre os dois tipos de dinâmicas, de acordo com a proposta da alteração da especificação no item b através de um *checkbox* denominado “Versão B”. Não selecionado o *checkbox,* a dinâmica é a descrita no item a. Selecionando o *checkbox* a dinâmica é a descrita pelo item b. Durante a execução, não é possível alternar entre as dinâmicas.

A seguir temos um quadro que mostra os itens descritos, assim como a sua disposição na interface. É importante lembrar que pequenas variações de layout podem decorrer em diferentes SOs.

Bibliografia

Referências bibliográficas utilizadas no estudo:

- Operating Systems: Internals and Design Principles (5th Edition)

ISBN-13: 978-0131479548

Apêndice

A seguir disponibilizamos o código fonte do Prog1 utilizado no estudo.

Método de execução

O Prog1 deve ser compilado e executado em um ambiente UNIX compatível através dos seguintes comandos:

```
A) Prog 1 - Este programa utiliza threads
gcc -o Prog1 Prog1.c -pthread -lm
./Prog1

B) Prog 1 - Este programa utiliza subprocessos
gcc -o Prog3c Prog3c.c -lm
./Prog3c
```

O Prog2 deve ser compilado e executado em qualquer ambiente que possua uma máquina virtual Java 1.6 ou superior.

```
Iniciar a IDE Eclipse
(No Linux LCI através de: eclipse-3.5 /opt/sun-jdk-1.6.0.26/bin/java)
  -Criar um Novo Projeto
  -Importar os arquivos do diretório ./Prog2

java -jar ./Prog2.jar
```