



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ

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

TEMA 1
APLICATIVO CLIENTE-SERVIDOR

Disciplina: Sistemas Operacionais
Professores: Thomé e Valéria

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

Sumário

Apresentação..... 3

 Especificações gerais.....3

 Modelagem.....4

 Versão 1..... 6

 Estrutura e Análise.....6

 Versão 2..... 7

 Estrutura e Análise.....7

 Interface e Operação.....8

Bibliografia..... 8

Apêndice..... 9

 Método de execução..... 9

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.

Modelagem

A Modelagem realizada do Sistema foi concebida baseando-se em nos conceitos adquiridos em sala de aula, na 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* Aterndentes 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.

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

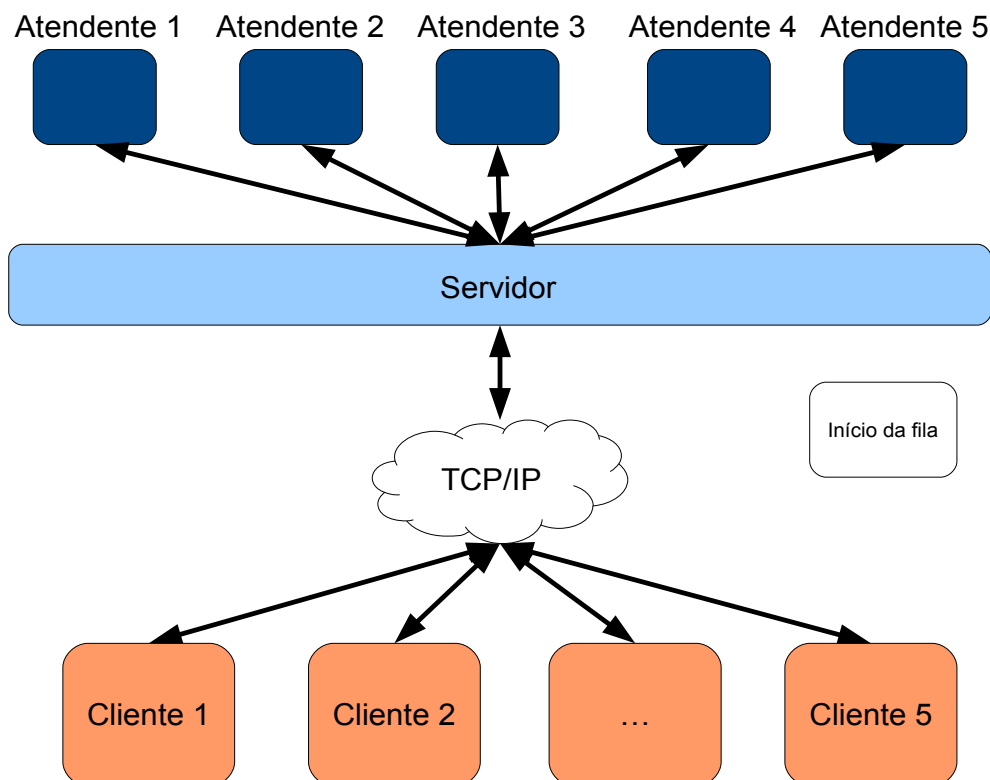
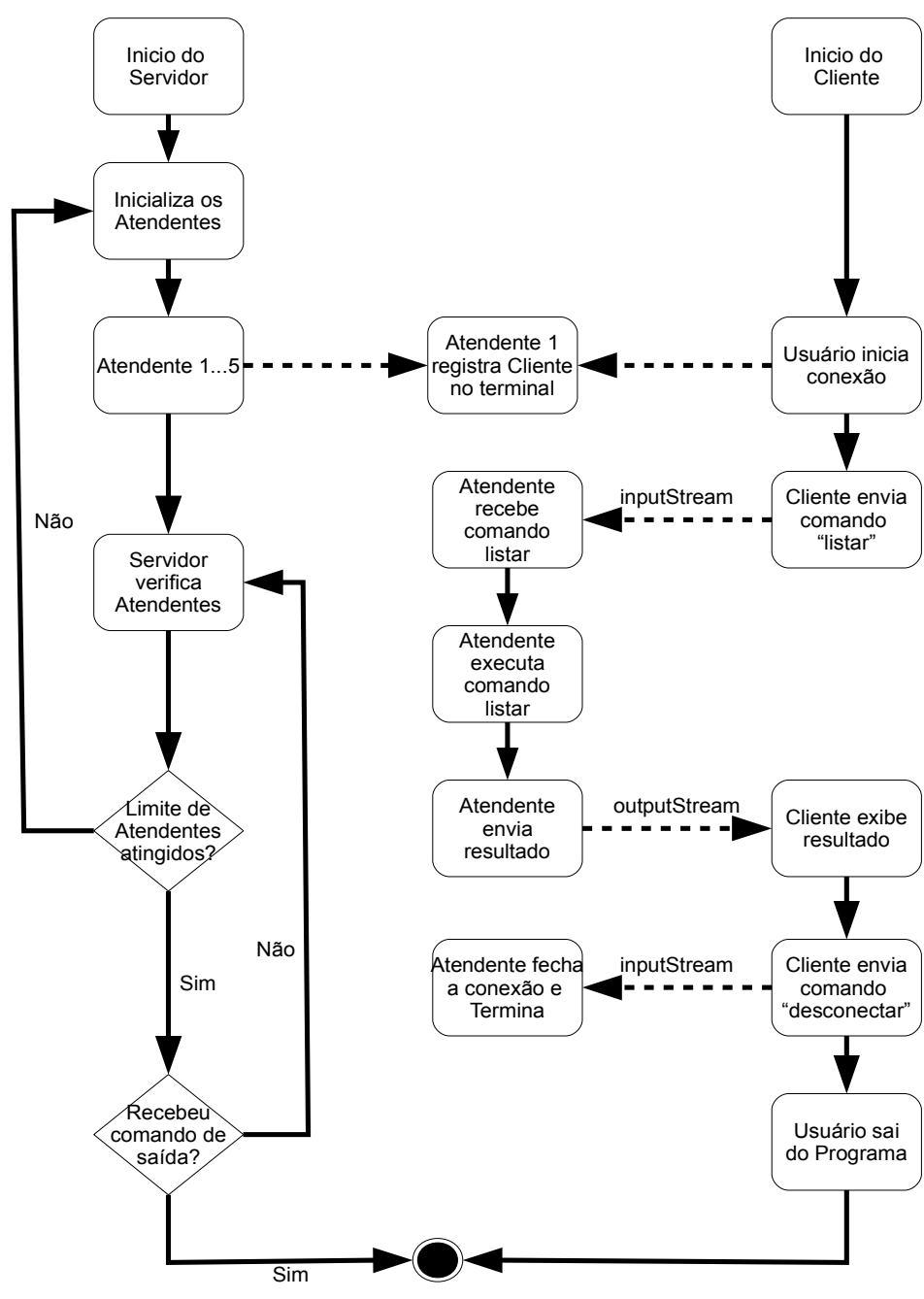


Figura 1

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



De acordo com o figura 1 temos o tratamento das requisições feitas através do *socket*. Desta forma o Servidor manipulafeito pelo Produtor/Consumidor. Neste problema temos m entidades chamadas de Produtor e n entidades chamadas de Consumidor. Também temos no sistema uma terceira entidade chamada de Recurso ou Produto. Nesse sistema, cada Produtor é responsável por produzir de forma independente das demais entidades, uma unidade do Recurso em tempo constante e finito. Da mesma forma, um Consumidor deve consumir uma unidade do Recurso também em tempo constante e finito.

A dinâmica deste sistema se dá pela forma em que os Produtores se organizam para produzir; como os Recursos são armazenados e disponibilizados aos Consumidores; e como estes se dispõem para consumir os Recursos.

Dadas as especificações do trabalho temos a seguinte dinâmica a ser implementada:

- 3 Produtores e 5 Consumidores.
- Produção de 1000 Recursos no total.
- Recursos dispostos em um fila (*buffer*) com 50 posições.
- Produtores inserem no final da fila.
- Consumidores consomem no início da fila.
- Ambiente assíncrono.

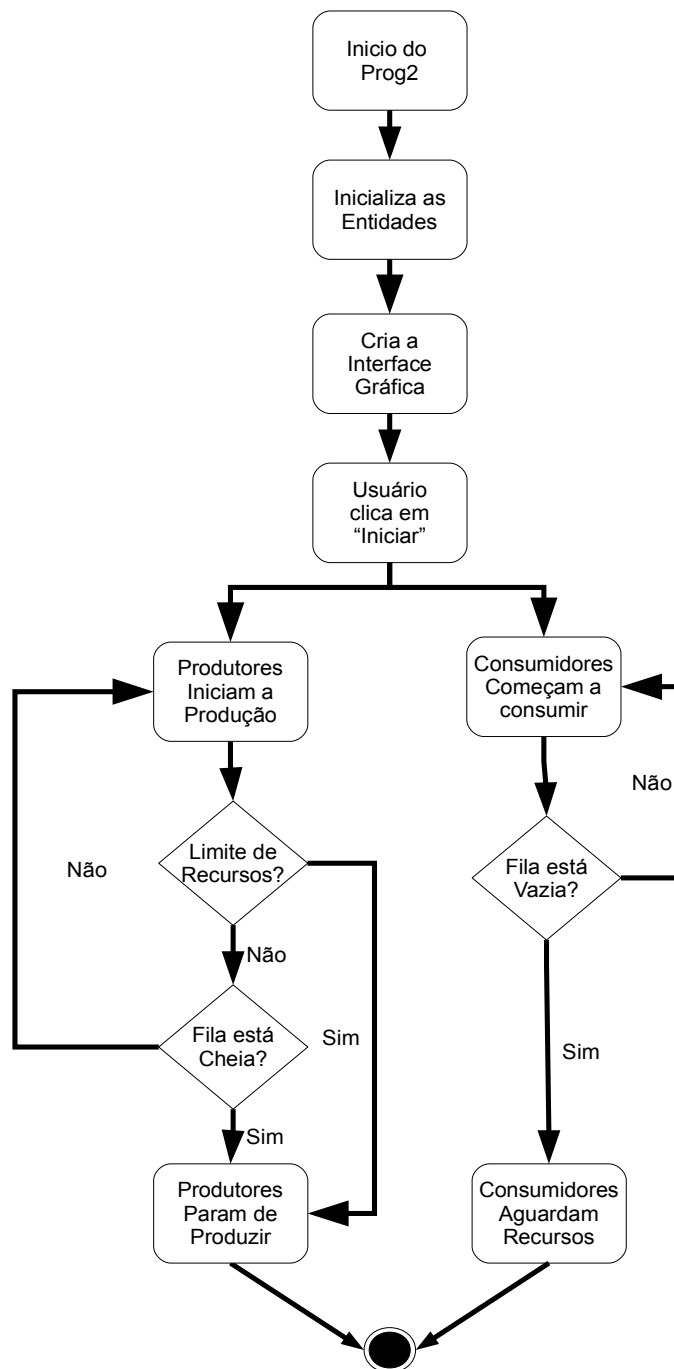
Para ilustrar em formas gerais como essa dinâmica se, dá podemos esboçar a seguinte figura:

A especificação do Prog2 também requeria que a simulação contasse com uma interface gráfica que exibisse os valores relacionados a implementação. Desta forma, diferente dos demais programas, foi escolhida a linguagem **Java** por possuir uma fácil implementação de Interface Gráfica além de compatibilidade entre os diversos SO.

Versão A

Estrutura e Análise

A versão A do Prog2 determina a execução da simulação através do fluxo de execução abaixo. Porém, é importante ressaltar que, dado um ambiente assíncrono, este fluxo tem como objetivo apenas exemplificar uma execução.



Quando as atividades do sistema Produtor/Consumidor são iniciadas pelo usuário, os produtores e consumidores disputam fatias de tempo no processador. De acordo com a especificação do trabalho, os Produtores tem a mesma prioridade de acesso ao processador do que os Consumidores. Nesta versão, os consumidores não disputam os recursos entre si, apenas seguem a fila circular.

Versão B

Estrutura e Análise

A versão B do Prog2 pode ser ilustrada através do mesmo fluxo de execução da versão A, já que este não representa a ordem de acesso dos Consumidores à fila de recursos.

Entretanto, nesta versão, como os Consumidores disputam os Recursos entre si, o que

pode ocorrer uma variação considerável do número de Recursos consumidos por cada Consumidor. Os Produtores também devem ter prioridade maior que os Consumidores no acesso a Região Crítica, acarretando em alguns períodos onde são produzidos diversos Recursos sem que os Consumidores tem acesso aos mesmos.

Interface e Operação

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
- Linux Man
 - http://linux.die.net/man/3/pthread_create
 - http://linux.die.net/man/3/pthread_join
 - http://linux.die.net/man/3/pthread_exit
 - http://linux.die.net/man/3/pthread_attr_init
 - http://linux.die.net/man/3/pthread_attr_setscope

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