

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN  
INSTITUTO METRÓPOLE DIGITAL - IMD  
SISTEMAS OPERACIONAIS

Relatório do projeto final da disciplina de Sistemas Operacionais

# Sistema de autenticação por identificação facial

**Breno Maurício de Freitas Viana**  
**Felipe Barbalho Rocha**

## **Resumo**

Este relatório apresenta o projeto final das disciplinas de Sistemas Operacionais e Laboratório de Sistemas Operacionais o qual foi denominado de Sistema de autenticação por identificação facial. A ideia deste projeto é de identificar rostos dos usuários a partir de imagens capturadas por uma câmera em tempo real e autenticar tais faces em um servidor, abordando os assuntos vistos em sala que foram implementados neste projeto.

Natal/RN,  
09 de junho de 2016

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Conceitos</b>	<b>3</b>
2.1	Thread . . . . .	3
2.2	Socket . . . . .	3
2.3	Manipulação de processos . . . . .	3
<b>3</b>	<b>Materiais utilizados</b>	<b>3</b>
<b>4</b>	<b>Implementação</b>	<b>4</b>
4.1	Cliente . . . . .	4
4.2	Servidor . . . . .	4
<b>5</b>	<b>Testes</b>	<b>5</b>
<b>6</b>	<b>Conclusão</b>	<b>6</b>
<b>7</b>	<b>Referências</b>	<b>7</b>

# 1 Introdução

Este relatório apresenta o projeto final das disciplinas de Projeto de Sistemas Operacionais (PSO) e de Laboratório Projeto de Sistemas Operacionais (LPSO) da Universidade Federal do Rio Grande do Norte (UFRN). O projeto desenvolvido trata de um sistema de identificação de faces que dispõe de um sistema cliente-servidor, onde o cliente é responsável por enviar as informações (as imagens) para autenticação no servidor.

Inicialmente, iremos apresentar os conceitos que foram dados nas disciplinas de PSO e LPSO que foram utilizados neste projeto. Em seguida, apresentaremos como foi implementado no projeto cada um desses conceitos, quais materiais (hardware) foram utilizados e como se deu a implementação do projeto. Por fim, mostraremos os resultados de alguns testes.

## 2 Conceitos

Os conceitos que foram dados em aula nas disciplinas de PSO e LPSO e que foram utilizados no desenvolvimento deste projeto são Threads, Sockets e Manipulação de processos. A seguir, temos mais detalhes da importância do uso e de como foram utilizados tais conceitos.

### 2.1 Thread

O uso de threads nos garante utilizar o real poder de processamento que o hardware oferece, e isso se deu no nosso projeto, proporcionando um grande ganho de performance na execução das aplicações. Como é um sistema cliente-servidor em tempo real temos que garantir uma certa velocidade no processo de autenticação que depende em maior parte no tempo de processamento de imagem realizado pelo cliente.

### 2.2 Socket

Como foi dito, utilizamos neste projeto um sistema de cliente-servidor, logo, o conceito de sockets foi imprescindível, no qual o seu uso fornece uma interface entre programas independentes através de um protocolo de comunicação possibilitando a concorrência das aplicações.

### 2.3 Manipulação de processos

O uso de manipulação de processos se deu com a configuração de Sinais no momento em que o programa chegasse ao fim. Isso é importante, porque muitas vezes um programa pode ser interrompido no meio de um processo, e para garantir que o ele seja finalizado é pode ser feito o uso de Sinal.

## 3 Materiais utilizados

Neste projeto utilizamos os seguintes materiais de hardware:

- BeagleBone Board;
- Câmera AXIS M1011;

- LEDs e resistores;

E os seguintes materiais de software:

- Sistemas Linux (Ubuntu e Debian);
- QT (IDE de desenvolvimento);
- OpenCV;
- Bitbucket (Repositório);

Utilizamos a BeagleBone Board como o cliente e um computador pessoal de um dos integrantes deste trabalho como servidor.

## 4 Implementação

Esta seção apresenta como se comporta o sistema cliente-servidor e como se deu a implementação o servidor e o cliente neste projeto.

### 4.1 Cliente

O cliente é responsável pela captura de vídeo, detecção facial e envio das imagens (com face) ao servidor. A captura de vídeo é feita utilizando uma câmera ou a própria webcam (depende da configuração no momento da execução).

O cliente consulta em tempo real o vídeo da câmera, e salva em um frame, um processo busca faces no frame e armazena uma lista de faces vetorizadas, um outro processo verifica se existe uma única face e se seu tamanho é ideal para o reconhecimento facial, em caso positivo, é criado um socket e a imagem capturada da face é enviada por buffer para o servidor, após esse processo é dado um tempo espera para que possa ser feita novas detecções. Caso o seja identificado mais de um rosto na tela ou um rosto não tenha as dimensões configuradas, uma mensagem de aviso é mostrada alertando o usuário, para que não haja problemas na autenticação.

Os LEDs no cliente simbolizam a catraca, ou seja, o controle da permissão de passagem de usuários. Onde as cores vermelho, amarelo e verde, significam o status de autenticação, onde significam, respectivamente, acesso não permitido, esperando autenticação ser concluída, e acesso permitido.

### 4.2 Servidor

O servidor tem a função de organizar as imagens salvas diferenciando de cadastro de usuários e autenticações feitas durante um dia em específico e mostra as autenticações que estão sendo feitas em tempo real, exibindo a foto do último cadastrado na tela, como pode ser visualizado na *Figura 1*.

As imagens são salvas no diretório `log_authentication` e tem o seguinte formato “YYYY-MM-dd.HH:MM:SS.username.png”. As informações da autenticação do usuário são salvas diretamente no nome do arquivo de imagem. Essas informações pode ser visualizado na *Figura 2* e *Figura 3*.

Na aba de “Config” é possível configurar o endereço IP do servidor e listar os endereços IP’s disponíveis, como pode ser visualizado na *Figura 4*.

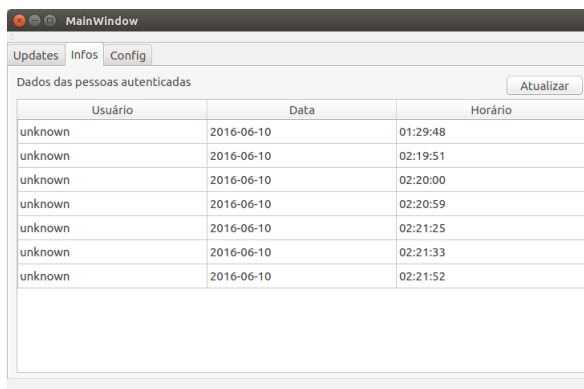


Figura 1: Tela de visualização para última face registrada

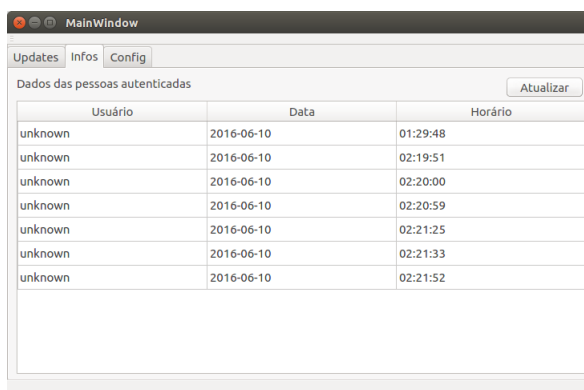


Figura 2: Tela de informações de registro



Figura 3: Diretório de registro de faces

O próximo passo para o desenvolvimento do projeto é o reconhecimento facial.

## 5 Testes

Primeiro, o servidor deve ser executado através do comando `./detecfaces-server`. Feito isso, para executar o cliente é necessário a informação do endereço do servidor, que pode ser visualizado nas configurações apresentado na *Figura 4*.

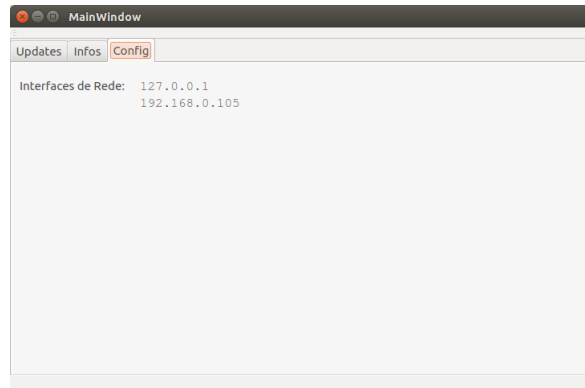


Figura 4: Tela de configurações do Servidor

Além do endereço do servidor, o cliente precisa fornecer o endereço da câmera, ou caso queira utilizar a webcam (caso possua) fornecer "0" como endereço da câmera. Por fim ainda há a opção de configurar o tempo de espera entre diferentes autenticações, e o tamanho mínimo da face que será autenticada.

Todas essas configurações são feitas como parâmetro na hora da execução do aplicação cliente, conforme a *Figura 5*.

```
Terminal File Edit View Search Terminal Help
breno@breno-PC:~/QTProjects/projeto_final/detectFace-client$ ./detectFace-client 0 192.168
.0.18
Parâmetros esperados: IP_CAM IP_SERV SLEEP_TIME SIZE_FACE_MIN
Por Padrão: IPCAM=0 IP_SERV=192.168.7.1 SLEEP_TIME=4 SIZE_FACE_MIN=200
-----
HIGHGUI ERROR: V4L/V4L2: VIDIOC_S_CROP
-----
IP SERVER      = 192.168.0.18
CAMERA IP      = 0
SLEEP TIME     = 4
SIZE_FACE_MIN  = 200
-----
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
Frame capturado
Análise de faces realizadas [resultado 0]
```

Figura 5: Tela de configurações do Servidor

Ainda na *Figura 5* é possível visualizar o log de informações que estão sendo executadas, como a captura de frames, e o resultado da busca por faces, outras variações de logs se dá quando duas faces são encontradas e quando uma face não atende ao tamanho mínimo.

Os resultados da execução do servidor pode ser visualizado nas Figuras 1, 2 e 4.

## 6 Conclusão

A partir do desenvolvimento do sistema de autenticação por identificação facial, nós pudemos aplicar conceitos vistos em sala de aula. Além disso, pudemos conhecer novas tecnologias, e obter um olhar crítico de sistemas atuais, formulando indagações como "Esse sistema poderia se tornar muito mais eficiente aplicando paralelismo de processos".

Por fim apesar de termos concluído nossas metas, muitas outras questões surgiram ao respeito do nosso sistema, entre elas, a aplicação de reconhecimento facial como chave única para cada autenticado, integração com outros sistemas de autenticação (autenticação por RFID), e questões que dizem respeito a confiança, integridade e segurança das informações com as quais manipulamos.

## 7 Referências

OpenCV. OpenCV documentation. Disponível em: <http://opencv.org/>. Acesso em: 06 de maio de 2016.

Stack Overflow question. Getting the source address of an incoming socket connection. Disponível em: <http://stackoverflow.com/questions/2064636/getting-the-source-address-of-an-incoming-socket-connection>. Acesso em: 06 de junho de 2016.

Stack Overflow question. Get the IP address of the machine. Disponível em: <http://stackoverflow.com/questions/212528/get-the-ip-address-of-the-machine>. Acesso em: 05 de junho de 2016.