

Ponto de Controle 02

Sistemas Embarcados

Leonardo Brandão Borges de Freitas

RA: 14/0025197

Universidade de Brasília, Campus Gama

Brasília, Brasil

leonardobbfga@gmail.com

Ygor Pereira Borgonove

RA: 14/0166408

Universidade de Brasília, Campus Gama

Brasília Brasil

ygor.borgonove@hotmail.com

Resumo – Desenvolvimento de um pequeno equipamento constituído por um display para gerar QR code, um HD para armazenamento de arquivos e um Raspberry Pi para embarcar procedimentos de upload e downloads de dados para uma nuvem física, offline e com alcance local de roteamento, em que o usuário terá acesso ao download através de senha e de upload dos arquivos se ocorrer a leitura do QR code.

Palavras-chave — Nuvem, Raspberry Pi, QR code, Download, Upload, Sistema Embarcado.

I. Introdução

Armazenamento e transferência de dados são palavras chaves em nosso cotidiano. Com o avanço da tecnologia digital, geramos cada vez mais material a ser compartilhado e distribuído. Para tanto, expandir a capacidade de servidores e acelerar o compartilhamento online de arquivos é foco de muitos desenvolvedores como a Google, Youtube, Amazon e vários outros.



Figura 1: Logo Google Drive

Em meio de tanta demanda se criou o serviço de nuvem, em que são fornecidos servidores para armazenamentos de banco dados, arquivos, backup, dentre outros serviços. Atualmente os maiores serviços de nuvem prestados são: Google Drive, Dropbox, One Drive e iCloud. Existem, basicamente, três tipos de núvens: IaaS, PaaS, SaaS.

IaaS é a mais básica das três em que o usuário aluga a infraestrutura de TI, servidores e máquinas virtuais, são as mais fornecidas pelas empresas citadas anteriormente.

PaaS, plataforma como serviço, é usada para serviços de computação em nuvem que fornecem um ambiente para desenvolvimento, testes, gerenciamento, dentre outras aplicações para empresas. Foi criada para auxiliar os desenvolvedores a criar aplicativos.

SaaS, software como serviço, basicamente é uma forma de distribuição e comercialização de software. Ao utilizar o SaaS as empresas contratantes não precisam se preocupar com instalação, manutenção e atualização de hardware ou software, só precisaria de uma conexão com a internet para gerenciar o que estivesse na nuvem. Muito utilizado para serviço de assinaturas.

II. Desenvolvimento

Em meio globalizado a segurança de arquivos está cada vez mais delicada e com alto custo. Muitas vezes nos deparamos com situações de vazamentos de arquivos pessoais em que o usuário paga caro por um produto ou serviço para armazenamento de seus materiais e devido a falhas na segurança, alguns mal intencionados conseguem ter acesso aos seus

arquivos pessoais podendo causar danos irreparáveis ao usuário. Um grande exemplo dessa quebra de segurança foi o que aconteceu com usuários da Apple, pessoas famosas tiveram fotos particulares expostas na internet recentemente. Para evitar problemas de quebra de segurança e vazamento de dados será construído uma nuvem offline que, para ter acesso, o usuário precisa estar conectado no Wi-Fi do Raspberry Pi 3, ou seja, o alcance máximo da nuvem é o próprio alcance do Wi-Fi integrado do Raspberry Pi 3.

Basicamente esta nuvem funcionará de dois modos. O primeiro modo será o de envio de arquivos, neste modo qualquer pessoa que possua a senha do Wi-Fi poderá submeter seus arquivos para a nuvem. Já no modo de download de arquivos além de possuir a senha do Wi-Fi o usuário precisará escanear um QR code que será gerado aleatoriamente e aparecerá em um display LCD de 7", ao escanear tal sQR code o usuário poderá fazer downloads dos arquivos armazenados previamente. Os arquivos que serão inseridos na nuvem vão ser armazenados em um cartão micro SD de 16 GB, caso esta memória for excedida será utilizado um HD externo com uma maior capacidade.

O OS que será utilizado neste projeto será o Raspian, foi escolhido este por ser simplificado e bem intuitivo, além de possuir uma interface gráfica bem interativa. Também será criada uma interface em forma de app em um celular android para que seja mais fácil e intuitivo enviar e fazer download de arquivos.

Tabela 1: Tabela de materiais

Material	Quantidade
Raspberry Pi 3	1
Display LCD touchscreen 7"	1
Jumpers	20
Cartão de micro SD 16 GB	1
Celular Samsung S3	1

III. Descrição de Software

Para que pudéssemos trabalhar com envios de arquivos para o Raspberry de uma forma wireless, foi feita uma configuração no Raspberry fazendo com este se tornasse um Wi-Fi access point, ou seja, possível acessar o Raspberry sem precisar conectar cabos. Essa configuração faz com que o Raspberry vire um roteador, assim, por meio de senha e usuário padrão o acesso remoto (via Wi-Fi) do Raspberry é liberado. Além disso o access point faz com que este seja um roteador de internet caso esteja conectado a um cabo ethernet. Para fazer esta configuração no Raspberry utilizamos um tutorial que está disponível no git [4].

Para trabalhar com envio de arquivos foi decidido que usaríamos a pasta Skullpi, esta pasta é volátil e funciona como a /tmp padrão do Linux, ou seja, após algumas manipulações dentro dela, os arquivos que estavam alocados dentro desta são apagados.

Foram desenvolvidos 3 scripts e um programa (em linguagem C) inicial para este ponto de controle. O primeiro se chama put.sh e pode ser visto abaixo.

```
#!/bin/bash
echo upload de arquivos
echo
echo compactando pasta
tar -zcf pac.tar Skullpi
echo
echo pasta compactada
echo
echo conectando ao rasp
sshpass -p "leonard7" scp pac.tar pi@192.168.42.1:/home/pi
echo upload feito com sucesso
echo apagando pac.tar
rm pac.tar
echo flush na skullpi
rm -r Skullpi
mkdir Skullpi
echo
```

Figura 2: put.sh

Esse script pega todos os arquivos que estão na pasta Skullpi do computador e compacta todos eles em um arquivo chamado pac.tar (linha 4). Após compactar os arquivos é feita a conexão com o Raspberry via ssh, em que é fornecida a senha "leonard7" e logo após é usada a função scp que transfere o arquivo pac.tar para a /home do Raspberry Pi. Após o envio a pasta pac.tar que está no computador é

apagada, é feito um flush na pasta Skullpi do computador. Após este flush a pasta Skullpi é apagada e logo após é criada novamente. Este processo ocorre para que esta pasta fique sempre limpa.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <pthread.h>
#include <signal.h>
#include <time.h>

int fd;

void tratamento_sinal(){
    puts("Fechando programa...");
    close(fd);
    exit(0);
}

int main(){
    char bff[8];
    signal(SIGINT, tratamento_sinal);

    while(1){
        system("./att.sh");

        if(fd = open("/home/pi/verificador.txt", O_CREAT | O_RDWR) < 0){
            puts("Erro na abertura do arquivo verificador");
            exit(0);
        }

        if(read(fd, &bff, sizeof(bff)) < 0){
            puts("erro na leitura do arquivo verificador.txt");
        }

        printf("%s\n", bff);
        sleep(1);

        if(!strcmp(bff, "pac.tar")){
            system("./descomp.sh");
        }

        close(fd);
    }

    return 0;
}
```

Figura 3: verificador.c

Este código feito em C é um verificador, ou seja, ele verifica se o arquivo pac.tar foi enviado corretamente para a /home, e caso esteja lá é feito o descompactamento da pasta pac.tar. Primeiramente este programa aciona outro script por meio da função system. Este script, att.sh, fica procurando na /home se existe o arquivo pac.tar utilizando a função grep, caso exista é escrito em um arquivo chamado verificador.txt a palavra “pac.tar”.

```
#!/bin/bash

echo verificando se pac esta na pasta
echo
ls /home/pi | grep pac.tar > /home/pi/verificador.txt
echo verificador att
echo
```

Figura 4: att.sh

Após a verificação o verificador.c abre o arquivo verificador.txt em modo de leitura e escrita ou caso o arquivo não esteja na pasta ele será criado. É feita a leitura deste arquivo e o conteúdo deste é alocado na variável bff que é um char. Após a leitura o programa compara, utilizando a função strcmp (biblioteca string.h), se a variável bff é igual a “pac.tar”, caso sejam diferentes o programa continua neste while infinito até achar um pac.tar. Caso sejam iguais é executado outro script chamado descomp.sh.

```
#!/bin/bash

echo Descompactar e excluir compactado
tar -zxvf /home/pi/pac.tar
echo arquivo descompactado
rm /home/pi/pac.tar
echo pac excluido
echo
```

Figura 5: descomp.sh

Ao entrar neste script o arquivo é descompactado e todos os arquivos que estavam dentro de pac.tar são armazenados em uma pasta chamada Skullpi que está dentro da /home do Raspberry. Depois de armazenar os arquivos na Skullpi o pac.tar é apagado da /home do Raspberry.

No código em C foi feito um tratamento de sinal que, quando o usuário aperta ctrl+c é mostrado na tela “Fechando o programa...” e este é imediatamente finalizado

Foram utilizadas algumas bibliotecas para a execução do código em C por exemplo: para executar a função system é necessário da biblioteca sys/types.h, para fazer o tratamento de sinal foi utilizada a biblioteca signal.h. Para fazer manipulações com arquivos (posix) são utilizadas duas bibliotecas: unistd.h e fcntl.h.

IV. Descrição de Hardware

Para fazer esta implementação será utilizado um Raspberry Pi 3 Model B. Foi escolhido este modelo pois ele possui Wi-Fi e bluetooth integrado. Além de um processador de 1.2 GHz de processamento, este modelo possui 1 GB de memória RAM, 40 portas GPIO, 4 portas USB e porta HDMI [3].

Referências

- [1]<https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>
- [2] <https://www.salesforce.com/br/saas/>
- [3]<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [4]https://github.com/leobbf/Sistemas_Embarcados/blob/master/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf

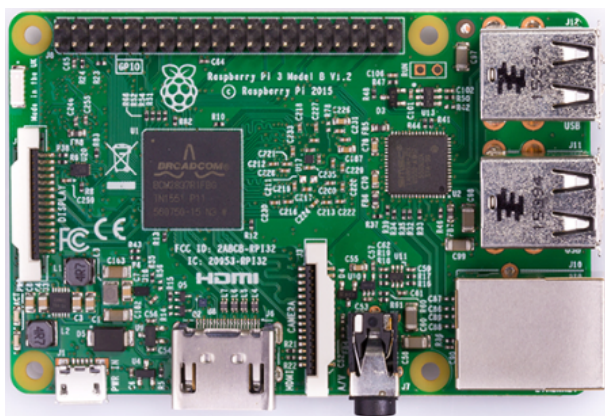


Figura 6: Raspberry Pi 3 Model B

Para mostrar o QR code ao usuário que deseja fazer downloads de arquivos será usado um display LCD touch screen da própria Raspberry. O QR code funcionará num processo semelhante ao do whatsapp web para autenticação.



Figura 7: Display LCD Touchscreen 800 x 480

V. Conclusão

Conclui-se que o projeto está encaminhado e que faltam somente algumas coisas para serem desenvolvidas. Entre essas coisas estão: um script pull, em que o Raspberry envia arquivos para outro aparelho, uma interface gráfica para computador e/ou um aplicativo para celular para que essa transferência seja mais intuitiva possível e por fim, um gerador de QR code que auxiliará na autenticação do usuário que deseja fazer download de arquivos.