

## TRABALHO INDIVIDUAL # 2 APLAY

Vilmey Francisco Romano Filho - 11/0021380

Programa de Graduação em Engenharia Eletrônica, Faculdade Gama  
Universidade de Brasília  
Gama, DF, Brasil  
email: vilmeyr@gmail.com

### 1. OBJETIVO

Este exercício pratico tem como objetivo colocar em prática a abertura e leitura de arquivos assim como a criação de pipes, para a comunicação entre diversas aplicações presentes no sistema operacional.

### 2. INTRODUÇÃO

Elaboração de um programa em linguagem C capaz de executar arquivos música não compactados, no formato WAV com a utilização da aplicação nativa do Linux APLAY. Este programa deve ser capaz de ser executado na plataforma Raspberry PI.

### 3. ESPECIFICAÇÃO

#### 3.1. Listagem de media

A primeira etapa deste programa consiste em listar todas as medias .wav do diretório local. E para isso geramos um arquivo com todas as musicas com o comando `system("ls — grep wav & media.log")`. Após a criação do log de musicas abrimos o arquivo para leitura e armazenamos os nomes das mídias em um array de strings (`musicas[][]`).

De posse dos nomes das musicas, pedimos para o usuário que escolha um das opções listadas, esta opção será gravada na string (`musica[]`).

#### 3.2. Abertura da mídia

O próximo passo é a abertura da música para leitura, e a abertura do pipe de comunicação para escrita. De posse dos descritores do arquivo de audio e do pipe stream podemos começar a execução do arquivo.

#### 3.3. Leitura e execução

A função `fread()` foi utilizada como critério de parada do laço de repetição `while()`. A função `fread()` retorna 1 quando no fi-

nal do arquivo.

O audio é lido por meio da função `fread()` para o buffer, que é consequentemente escrito no pipe `write()`.

Quando a execução do arquivo é finalizada fechamos a comunicação com o pipe e fechamos também o arquivo de audio.

### 4. CONCLUSÕES

O experimento ocorreu de acordo com o esperado. Proporcionou uma melhor visualização do papel dos pipes, e ofereceu novas utilidades aos programas em C quando combinados com os aplicativos presentes no sistema operacional.

## 5. CÓDIGO

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <signal.h>
5 #include <string.h>
6 #include <errno.h>
7
8 //Funcao para listar arquivos no diretorio atual,
9 //recebe o ponteiro para musica
10 void Listar(char *musica)
11 {
12     int i=0, j=0, numero;
13     char buffer[30], musicas[30][30];
14     FILE *media;
15
16     //Lista as musicas .wav para media.log
17     system("ls | grep wav > media.log");
18
19     media = fopen("media.log", "r");
20
21     while(1)
22     {
23         int ret = fscanf(media, "%s\n", buffer);
24         if(errno != 0)
25         {
26             perror("scanf:");
27             break;
28         }
29
30         //Fim do arquivo, sair do loop
31         else if (ret == EOF)
32             break;
33
34         strcpy(musicas[i], buffer);
35         //memcpy(musicas[i], buffer, strlen(buffer));
36         i++;
37     }
38
39     fclose(media);
40
41     //Printa as opcoes de musica
42     for(j=0; j<i; j++)
43         printf("Musica[%d]: %s\n", j, musicas[j]);
44
45     printf("Digite o numero da musica: ");
46     scanf("%d", &numero);
47
48     //Passamos a musica selecionada para a variavel
49     //musica
50     strcpy(musica, musicas[numero]);
51 }
52
53
54
55 int main()
56 {
57     int i, fd, size =0;
58     char buffer[256], musica[256];
59
60     FILE *audio_fd;
61     FILE *stream;
62
63 }
```

```
64 while(1)
65 {
66     //Passa um ponteiro de char para receber o
67     //nome da musica
68     Listar(musica);
69
70     //Descritor para o arquivo de musica
71     audio_fd = fopen(musica, "r");
72     if(!audio_fd)
73     {
74         printf("Erro na abertura fopem");
75     }
76
77     //Abre um pipe entre o programa e o aplay
78     stream = popen("aplay -t wav -", "w");
79     if(!stream)
80     {
81         printf("Erro na abertura do pipe\n");
82     }
83
84     //Fileno retorna um descritor da stream no
85     //formato int.
86     fd = fileno(stream);
87
88     //Executa arquivo ate o FEOF
89     while(!feof(audio_fd))
90     {
91         fread(buffer, sizeof(buffer), 1, audio_fd);
92         write(fd, buffer, sizeof(buffer));
93     }
94
95     //Fecha o pipe e o arquivo de audio
96     pclose(stream);
97     fclose(audio_fd);
98 }
99
100 return 0;
101 }
```

Codigo/aplay\_codigo