

Prática 3 - Sistemas Operacionais

Elton Mauricio Da Silva - RA 11201810955

link de arquivos: <https://github.com/eltonmds/ufabc_operational_systems>

1 Questao 1

A solução proposta utiliza o método ‘named pipe’ onde há um processo responsável por criar um arquivo através da função `fifo`, o processo coletor e o displayer se comunicam através desse arquivo. Segue abaixo o código em C.

named_pipe

Processo responsável por criar o arquivo utilizando a função `fifo`.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/stat.h>
4  #include <sys/types.h>
5
6
7  int main() {
8      const char* pipeF = "fifoF";
9
10     if (mkfifo(pipeF, 0777) < 0) {
11         printf("Ops... erro:");
12         exit(1);
13     }
14     else {
15         printf("Queue named pipe feita!!!\n");
16     }
17 }
```

coletor

O processo coletor é responsável por escrever as cotações no arquivo `fifo`

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <sys/stat.h>
5  #include <sys/types.h>
6  #include <unistd.h>
7  #include <stdlib.h>
8
9  #define FIFO_FILE "fifoF"
10
11 struct cotacao {
12     char moeda_origem[4];
13     char moeda_destino[4];
```

```

14     float valor;
15     char data_hora[20];
16 };
17
18 int main() {
19     int fd;
20     struct cotacao cotacoes[15] = {
21         {"USD", "BRL", 5.25, "2023-01-20T10:31:00"},
22         {"EUR", "BRL", 6.20, "2023-01-20T10:31:00"},
23         {"CNY", "BRL", 0.76, "2023-01-20T10:31:00"},
24         {"RUB", "BRL", 0.07, "2023-01-20T10:31:00"},
25         {"ARS", "BRL", 0.03, "2023-01-20T10:31:00"},
26         {"ASA", "BRL", 0.30, "2023-01-20T10:31:00"},
27         {"ABL", "BRL", 0.35, "2023-01-20T10:31:00"},
28         {"SFR", "BRL", 5.35, "2023-01-20T10:31:00"},
29         {"LEV", "BRL", 1.35, "2023-01-20T10:31:00"},
30         {"BAR", "BRL", 3.35, "2023-01-20T10:31:00"},
31         {"THS", "BRL", 2.35, "2023-01-20T10:31:00"},
32         {"BTC", "BRL", 145000.25, "2023-01-20T10:31:00"},
33         {"ETH", "BRL", 9430.35, "2023-01-20T10:31:00"},
34         {"ADA", "BRL", 1.55, "2023-01-20T10:31:00"},
35         {"AJA", "BRL", 2.35, "2023-01-20T10:31:00"},
36         {"BOC", "BRL", 0.75, "2023-01-20T10:31:00"},
37     };
38
39     fd = open(FIFO_FILE, O_WRONLY);
40
41     if (fd < 0) {
42         perror("open");
43         exit(1);
44     }
45
46     for (int i = 0; i < 15; i++) {
47         write(fd, &cotacoes[i], sizeof(struct cotacao));
48     }
49
50     close(fd);
51
52     return 0;
53 }

```

displayer

o processo `displayer` lê as informações escritas pelo coletor no arquivo `fifo` e apresenta as informações.

```

1  #include <stdio.h>

```

```
2  #include <fcntl.h>
3  #include <sys/stat.h>
4  #include <sys/types.h>
5  #include <unistd.h>
6
7  #define FIFO_FILE "fifoF"
8
9  struct cotacao {
10     char moeda_origem[4];
11     char moeda_destino[4];
12     float valor;
13     char data_hora[20];
14 };
15
16 int main() {
17     int fd;
18     struct cotacao cotacao;
19
20     fd = open(FIFO_FILE, O_RDONLY);
21
22     while (read(fd, &cotacao, sizeof(struct cotacao)) > 0) {
23         printf("Moeda origem: %s\n", cotacao.moeda_origem);
24         printf("Moeda destino: %s\n", cotacao.moeda_destino);
25         printf("Valor: %.2f\n", cotacao.valor);
26         printf("Data-hora: %s\n", cotacao.data_hora);
27         printf("\n");
28     }
29
30     close(fd);
31
32     return 0;
33 }
```