

Aquele Flau - Sinal Reverso

Gean é um jovem rapaz que controla as câmeras do *flow podcast*. Esse controle é feito apertando botões em um teclado especial. Apesar de ser um podcast, o programa é transmitido ao vivo por vídeo e imagem, e está bem famoso.

O cenário é composto por 4 microfones e 5 câmeras. Os microfones ficam sempre ligados captando a voz das pessoas na mesa, em geral temos *Igor3k* e *Monark* na mesa, que são os apresentadores, e do outro lado temos uma ou duas pessoas, que são as convidadas. As câmeras apontam para cada um dos 4 lugares da mesa e ainda temos uma câmera de visão geral, que mostra todo mundo.

Para chavear entre as câmeras, o operador Gean, pressiona um botão em seu teclado especial de controle, por exemplo, u_1 manda para a câmera do *Igor3k*, u_2 para o *Monark*, u_3 convidado 1, u_4 convidado 2 e por fim u_5 para a visão geral.

Em momentos acalorados fica difícil de acompanhar e mandar a câmera para a pessoa certa. Por isso *Aquele Flau* te contratou para escrever um programa de computador para ajudar no chaveamento de câmera. A ideia é sempre acertar a câmera para a pessoa que está falando, e quando estiver em silêncio abrir para a câmera de visão geral.

No atual momento você já possui um controlador que identifica a voz de quem está falando (o algoritmo foi treinado, já funciona e você será um usuário dele :)), a placa já consegue capturar a informação de quem está falando, mas ainda é lenta e estamos validando todo o ambiente. Para tanto, temos que entender algumas coisas.

Sempre que a câmera é trocada pelo teclado especial, um sinal é gerado e seu processo recebe um sinal (gerado de forma parecida com a chamada de sistema `kill(2)`), os sinais são mapeados da seguinte forma:

- SIGUSR1 - câmera do *Igor3k*
- SIGUSR2 - câmera do *Monark*
- SIGINT - câmera do Convidado 1
- SIGTERM - câmera do Convidado 2
- SIGALRM - câmera geral

Sempre que seu processo receber um sinal (mas somente quando receber um sinal) você deverá ler uma única linha da entrada padrão. A linha na entrada padrão possuirá uma única string `s` de comprimento máximo de 10 caracteres, podendo ser:

- Igor3k
- Monark
- Con1

- Con2
- Silencio

O seu objetivo é validar a troca de câmera com o sinal recebido, ou seja, se você receber o sinal `SIGUSR1` e ler da entrada padrão `Igor3k` é um acerto, e você deve imprimir na saída padrão uma única linha contendo a frase `Certo`. Quando a câmera apontar para uma pessoa diferente da que foi lida na entrada padrão você deve imprimir na tela `Erro` e enviar um sinal `SIGALRM` para o processo pai.

Seu programa deve encerrar assim que receber um sinal de câmera, e o `scanf` retornar EOF.

DICA

A principal lógica do seu programa acontecerá dentro da função que trata sinais. A função principal irá apenas instalar o tratador de sinais e ficará esperando eternamente.

O seu programa somente irá encerrar quando receber um sinal e a entrada padrão não possuir mais dados para serem lidos. Logo sempre teremos um sinal a mais que do que linhas da entrada padrão.

Exemplo 1

Supondo a seguinte ordem de sinais:

```
SIGUSR1
SIGUSR2
SIGALRM
```

E a entrada padrão seguir a ordem:

```
Igor3k
Monark
```

O seu programa deve imprimir a seguinte ordem:

```
Certo
Certo
```

Exemplo 2

Supondo a seguinte ordem de sinais:

```
SIGINT
SIGTERM
SIGUSR1
SIGTERM
```

```
SIGUSR2
SIGALRM
SIGINT
```

E a entrada padrão seguir a ordem:

```
Con1
Con2
Igor3k
Monark
Monark
Silencio
```

O seu programa deve imprimir a seguinte ordem:

```
Certo
Certo
Certo
Erro
Certo
Certo
```

- quando imprimiu `Erro` também deve ter enviado um sinal `SIGALRM` para o processo pai

Exemplo 3

Supondo a seguinte ordem de sinais:

```
SIGUSR1
SIGUSR2
SIGALRM
SIGALRM
SIGTERM
```

E a entrada padrão seguir a ordem:

```
Monark
Igor3k
Con1
Silencio
```

O seu programa deve imprimir a seguinte ordem:

```
Erro
Erro
```

Erro
Certo

- Para cada vez que imprimiu `Erro` também deve ter enviado um sinal `SIGALRM` para o processo pai

Author: Bruno Ribas