

Disciplina: Sistema Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 11_GPIO_2

1. Escreva um código em C para gerar uma onda quadrada de 1 Hz em um pino GPIO do Raspberry Pi.

```
#include "gpio_sysfs.h"
#include <unistd.h>
//Compilar junto com o sysfs.c
int main()
{
    int pin=18;
    if(setGPIO_Out(pin))
        return -1;
    if (GPIO_Write(pin,1))
        return 1;
    sleep(0.5);
    if(unsetGPIO(pin))
        return 2;
    return 0;
}
```

2. Generalize o código acima para qualquer frequência possível.

```
#include "gpio_sysfs.h"
#include <unistd.h>
//Compilar junto com o sysfs.c
int main()
{
    int pin=18;
    float f = 0, t = 0;

    printf("freq:\n");
    scanf("%f", &f);

    t = (1/(2*f));

    if(setGPIO_Out(pin))
        return -1;
    if (GPIO_Write(pin,1))
        return 1;
```

```
sleep(t);  
if(unsetGPIO(pin))  
    return 2;  
return 0;  
}
```

3. Crie dois processos, e faça com que o processo-filho gere uma onda quadrada, enquanto o processo-pai lê um botão no GPIO, aumentando a frequência da onda sempre que o botão for pressionado. A frequência da onda quadrada deve começar em 1 Hz, e dobrar cada vez que o botão for pressionado. A frequência máxima é de 64 Hz, devendo retornar a 1 Hz se o botão for pressionado novamente.

```
#include <unistd.h>  
#include <fcntl.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <signal.h>  
#include <time.h>  
#include <sys/types.h>  
  
int fd;  
int fp[2];  
  
void fechar(){  
    close(fd);  
    fd = open("/sys/class/gpio/unexport",O_WRONLY);  
    write(fd,"20",2);  
    write(fd,"21",2);  
    close(fd);  
    printf("Fechando programa\n");  
    sleep(1);  
    exit(0);  
}  
  
int main(){  
  
    pid_t pid_id;  
    pipe(fp);  
    pid_id = fork();  
    signal(SIGINT,fechar);  
  
    if(pid_id == 0){  
  
        char bf='0';  
        int frequencia = 1;
```

```
//Setando como export
printf("Realizando o export 20\n");
fd = open("/sys/class/gpio/export",O_WRONLY);
write(fd,"20",2);
close(fd);

//Setando como saída
printf("Iniciando o pin 20 como saída\n");
fd = open("/sys/class/gpio/gpio20/direction",O_WRONLY);
write(fd,"out",4);
close(fd);

fd = open("/sys/class/gpio/gpio20/value",O_WRONLY);
printf("Iniciando o blink em 20\n");
while(1){
    printf("STATUS = %d\n",read(fp[0],&bf,sizeof(bf)));
    if(bf == '1'){
        frequencia = frequencia*2;
        bf = 0;
    }
    printf("Frequencia = %d BF = %c\n",frequencia,bf);
    write(fd,"1",2);
    usleep(500000/frequencia);
    write(fd,"0",2);
    usleep(500000/frequencia);
}

}

else{

//processo pai
char btn;

//Setando como export
printf("Realizando o export 21\n");
fd = open("/sys/class/gpio/export",O_WRONLY);
write(fd,"21",2);
close(fd);

//Setando como saída
printf("Iniciando o pin como saída 21\n");
fd = open("/sys/class/gpio/gpio21/direction",O_WRONLY);
write(fd,"in",4);
close(fd);
```

```
fd = open("/sys/class/gpio/gpio21/value",O_RDWR);
printf("Pronto para capturar gpio21\n");
while(1){
    lseek(fd,0,SEEK_SET);
    read(fd,&btn,2);
    printf("BTN = %c\n",btn);
    if(btn == '1'){
        write(fp[1],&btn,sizeof(btn));
    }
    usleep(500000);
}

}

return 0;

}
```