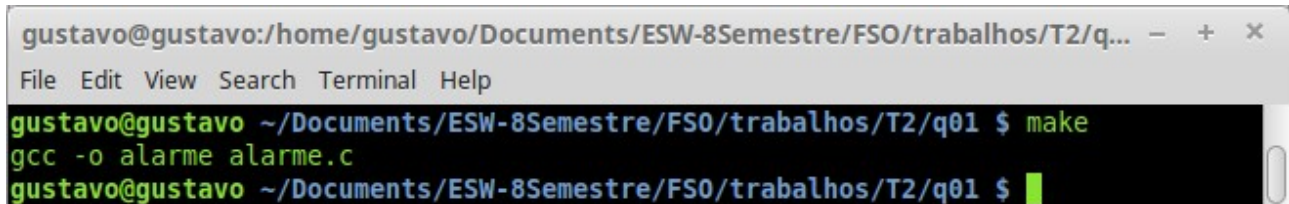


Questão 01 do T2 de Fundamentos de Sistemas Operacionais.

1 – Instruções

1.1 – Compilação:

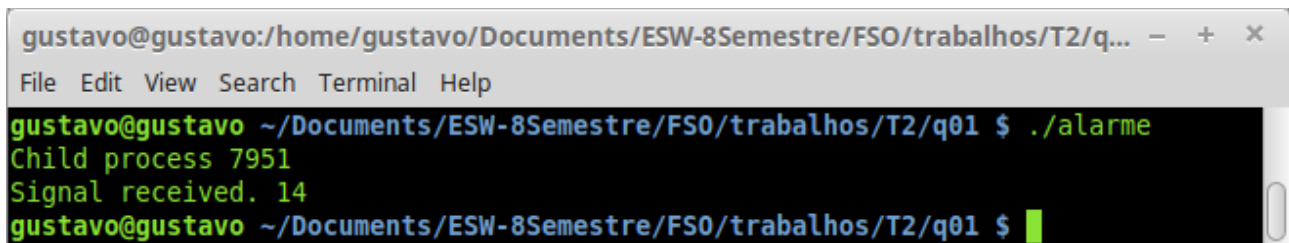
Para gerar o programa executável, escreva o comando 'make' no diretório da questão 02.

A terminal window with a title bar showing the path 'gustavo@gustavo:/home/gustavo/Documents/ESW-8Semestre/FSO/trabalhos/T2/q...'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the user running 'make' in the directory '~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01'. The output is 'gcc -o alarme alarme.c', followed by a new prompt line.

```
gustavo@gustavo:/home/gustavo/Documents/ESW-8Semestre/FSO/trabalhos/T2/q... - + x
File Edit View Search Terminal Help
gustavo@gustavo ~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01 $ make
gcc -o alarme alarme.c
gustavo@gustavo ~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01 $
```

1.2 – Executável:

Para executar o programa, basta digitar ./alarme na linha de comando dentro do diretório da questão 01.

A terminal window with the same title bar and menu bar as the previous screenshot. The terminal content shows the user running './alarme' in the directory '~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01'. The output is 'Child process 7951' and 'Signal received. 14', followed by a new prompt line.

```
gustavo@gustavo:/home/gustavo/Documents/ESW-8Semestre/FSO/trabalhos/T2/q... - + x
File Edit View Search Terminal Help
gustavo@gustavo ~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01 $ ./alarme
Child process 7951
Signal received. 14
gustavo@gustavo ~/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01 $
```

Irá ser impresso o número PID do processo filho, e depois de 5 segundos o número referente ao signal SIGALRM.

2 – Código fonte.

O código fonte contém explicações sobre o que cada uma das funções fazem em forma de comentário.

```
gustavo@gustavo:/home/gustavo/Documents/ESW-8Semestre/FSO/trabalhos/T2/q01
File Edit View Search Terminal Help
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

void signal_handler(int mysignal){
    printf("Signal received. %d\n", mysignal); // Emite uma mensagem com o código do sinal.
}

int main(int argc, char *argv[]){
    pid_t child_pid = fork(); // Faz uma cópia do processo.

    if ( child_pid == 0 ){ // Processo filho.
        printf("Child process %d\n", getpid()); // Printa na tela o valor PID do processo filho.
        sleep(5); // Faz o processo filho dormir por 5 segundos.
        kill(getppid(), SIGALRM); // Envia um sinal do tipo SIGALRM para o processo pai
                                // através da função getppid(), no qual retorna o valor PID
                                // do processo pai.
    }else if ( child_pid > 0 ){ // Processo pai.
        signal(SIGALRM, signal_handler); // Trata o recebimento de um sinal.
        pause(); // Deixa o processo pai rodando até receber um sinal.
    }else{
        printf("Fork failed!\n");
        return 1;
    }

    return 0;
}
```

35,0-1 All