

```
#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <time.h>

#include <stdlib.h>

#include <dirent.h>

#include <stdio.h>

#include <string.h>

#include <getopt.h>

#include <stdbool.h>

#include <ctype.h>

#include <sys/wait.h>

#include <signal.h>

#include <sys/mman.h>

#include <sys/time.h>
```

```
#define INTERVAL 2
```

```
int t=0;
```

```
void display_message()
```

```
{
```

```
    kill(0,SIGKILL);
```

```
    t=1;
```

```
}
```

```
void calling2()
```

```
{
```

```
    sleep(3);
```

```
}
```

```
void calling()
```

```
{
```

```
    signal(SIGALRM, display_message);
```

```
    int i;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        pid_t pID = fork();
```

```
        if (pID == 0)          // child
```

```
        {
```

```
            calling2();
```

```
            if(t==1)
```

```
            {
```

```
                printf("killing the process");
```

```
                kill(0,SIGKILL);
```

```
                exit(0);
```

```
            }
```

```
            exit(0);
```

```
        }
```

```
    else if(pid>0)
    {
        if(t==1)
        {
            printf("killing the process");

            kill(0,SIGKILL);

            exit(0);
        }

        printf("\nhello from the father");
    }
}
```

```
int main()
{
    signal(SIGALRM, display_message);

    alarm(2);

    calling();

    // wait until alarm callback before terminating main thread

    sleep(15);
}
```

Output

hello from the father

hello from the father