Signals

Objective: In this lab you will create an LKM called "signalex". The LKM send signals from kernel mode to a user mode process called "siguser" which will be provided.

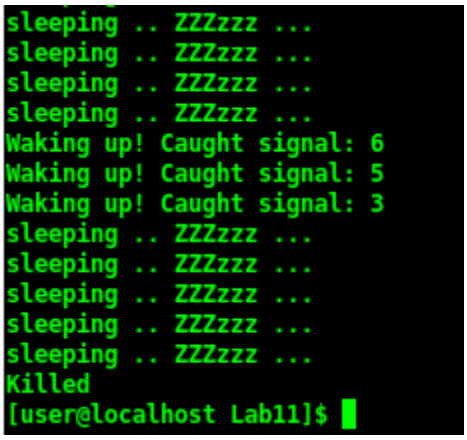
File(s) for this lab:

1. The "siguser" process is capable of handling the signals SIGABRT, SIGTRAP, and SIGQUIT. The code for the user mode program is below:

```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>

void sigcatch(int sig)
{
	printf("Waking up! Caught signal: %d\n", sig);
}

int main(void)
{
	struct sigaction act;
	memset(&act, 0, sizeof(act));
	act.sa_handler = sigcatch;

	sigaction(SIGABRT, &act, 0);
	sigaction(SIGTRAP, &act, 0);
	sigaction(SIGQUIT, &act, 0);

	while(1)
	{
		printf("sleeping .. ZZZzzz ...\n");
		sleep(1);
	}
}
```

2. Your LKM “signalex” should send those signals to the user mode process using the send_sig_info() API call.

3. Finally, your LKM should pause for 5 seconds to allow those signals to process out of the signal queue before sending a signal using the force_sig() function.



- Create a new project “Lab11” and import the existing code from “LKI/Lab11”.