CS 3305B

Intro to Signals

Lecture 6

Jan 25 2017

Introduction

- A signal is a mechanism for notifying a process that an event has occurred.
  - When a signal is sent to a process is normal execution is interrupted
- Events can arise from executing an instruction in the process's instruction stream
  - Illegal instruction e.g., divide by zero
  - Illegal address e.g., accessing A[11] when there is no A[11]

Even ilegal instructions send signals,
when they receive signals they must take actions

Introduction

kill -l cmd in linux will give you all the different signals types

- Events occur at any time and come from an external source
  - may be unrelated to the execution of the process
  - e.g., ctrl-D, ctrl-C, ctrl-Z

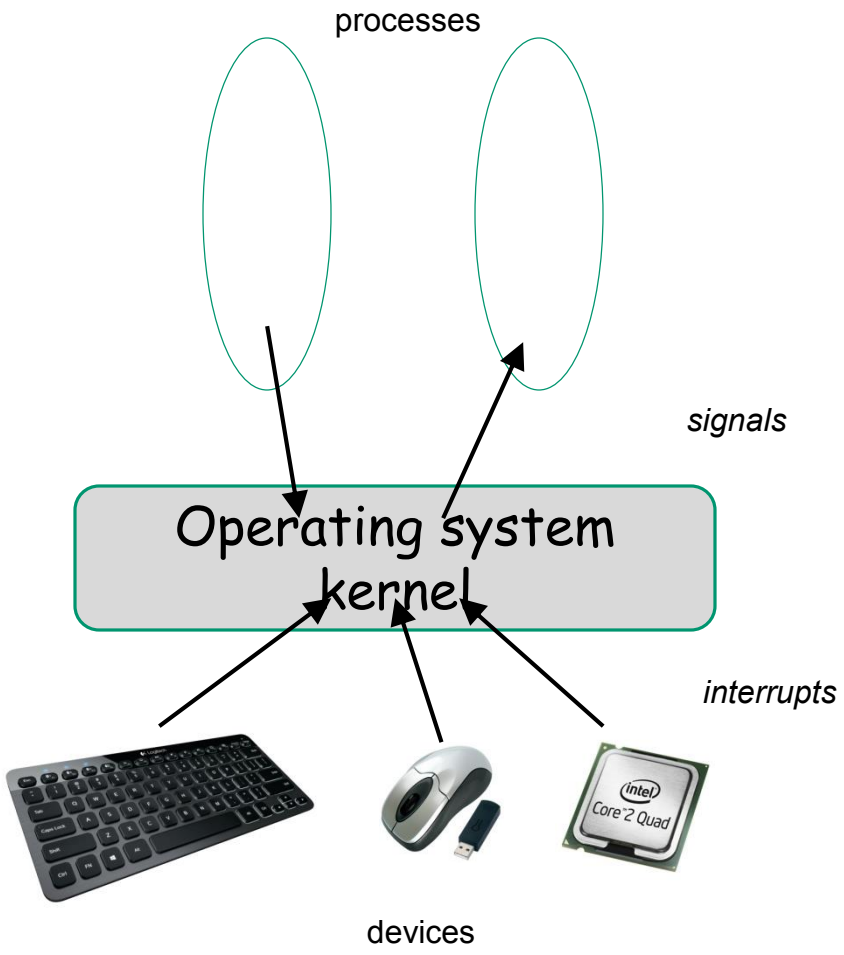
- Upon receipt of a signal a process may take some action

already defined by your operating system

  - Take a default action; or
  - Use a pre-defined signal handler

Introduction

- Signal sending:
  - OS kernel updates info for destination process
- Signal receiving:
  - kernel forces target process to handle signal
- A process can block some signals



Dealing with Signals

- Each signal type has a system-defined default action.
  - abort and dump core (SIGSEGV, SIGBUS, etc.)
  - ignore, stop, exit, continue
- A process may choose to block or ignore some signal types.

Dealing with Signals:Actions

- There are different actions that a process may choose to deal with a signal
  - Ignore
    - Exceptions: SIGKILL and SIGSTOP
  - Default
    - Different for different signals
  - Programmer-specified handler
    - Used instead of default

Example

```
int alarmflag=0;
alarmHandler ()
{   printf("An alarm clock signal was received\n");
    alarmflag = 1;
 }
main()
{
    signal (SIGALRM, alarmHandler);
    alarm(3);
     printf("Alarm has been set\n");
    while (!alarmflag) pause ();
   printf("Back from alarm signal handler\n
}
```

Sets up signal handler

Instructs OS kernel to send SIGALRM in 3 seconds

Suspends caller until signal

Signal Handling

- The system call signal captures a specific function and associates it with a programmer-defined function
- To use the signal system call requires that you include signal.h
- The form of the signal system call does vary across different versions of Linux/Unix

Important Signals

- SIGINT
  - Interrupt signal from terminal (ctrl-c)
- SIGTSTP
  - Stop signal from terminal (ctrl-z)
- SIGCHLD
  - A child process has stopped or terminated