

---

# Don't do this at home

---

ID1206 OPERATING SYSTEMS



# Signals

---

- Notifications sent to a process to inform that a certain event has occurred
- Causes a process to stop executing and handle the signal that has been received
- Type in 'man 7 signal' in the shell to list all signals and their description

# Signals used in experiment

---

- SIGINT: interrupt a process from keyboard (Ctrl+ C)
- SIGTERM: terminate a process
- SIGKILL: kill process immediately
- SIGFPE: arithmetic exception (division by 0)
- SIGSEGV: invalid memory access (segmentation fault)

# Sending signals

---

- From the keyboard
  - Ctrl-C
- From the shell
  - `kill -signal pid` : sends the specified signal to the process with the specified process id
- Using system calls
  - `kill(pid, signal)`: sends the specified signal to the process with the specified process id

# Interrupt descriptor table

---

- Used to determine the correct response to interrupts and exceptions
- It contains pointers to different procedures that should be executed on different interrupts, each signal has its default action to be taken
- You can define your own signal handling procedures

# Signal handlers

---

- Each signal has a default action
  - Ex: SIGINT – terminate process
- The action can be changed by using the sigaction structure
- Define a function that is invoked when the signal is received
- The default actions for SIGKILL and SIGSTOP cannot be changed

# Sigaction

---

- sigaction(signal, action, oldaction)
  - signal – the signal we want to handle
  - action – the action that should be taken when receiving the signal
  - oldaction – needed if you want to see what the old action was,  
should be set to NULL if it is not needed

# Other structures

---

- `siginfo_t`
  - Structure that contains information about the signal
  - For ex: the process identifier
- `uncontext_t`
  - Save the context of the process that caused the fault
  - For ex: program counter, contains address of the next instruction to be executed