CMPN303 Operating Systems

Lab 4 Signals

What is Signal

- Notify a process of an important event
- It interrupts the process (Handled Immediately)
- Each Signal has a “Handler”

Example

- Kill a process (End process in windows)
  - SIGKILL or 9 (non-maskable)
- Stop Process (Suspend)
  - SIGSTOP (non-maskable)
- Continue Process
  - SIGCONT
- Interactive signals
  - Ctrl+c (SIGINT)
  - Ctrl+z (SIGSTP)
- There are others in “Signal.h” or using kill -l

User Defined Signal

- We have 2 User Defined Signals
- “SIGUSR1” & “SIGUSR2”
- you could define their behavior freely (default is terminate)

Try signal01.c

- Try to run “./signal01.out &”
- How to kill the lost process?

Try signal01.c

- Try to run “./signal01.out &”
- How to kill the lost process?

The Answer is …

Kill -9 PID

Try signal01.c

- Try to run “./signal01.out &”
- How to kill the lost process?

The Answer is ...

Killall -9 signal01.out

Signal Handler

- We said each Signal has a handler
- How to create a handler for a specific Signal?
- Check “Signal02.c”
- What will happen when we uncomment “signal(SIGINT, SIG_DFL);” ?
- What if we added signal(SIGINT, SIG_IGN)

Signal Handler

- Some Unix/Linux systems return the signal handler to default automatically
- So it is good to always add your handler again in the end of the handler code “signal(SIGINT, myHandler);”

SIGCHLD

- When a Child Exit (interrupted, or resumes after being interrupted) it sends a “SIGCHLD” signal to it’s parent.
- Try “Signal03.c”

How do we send a Signal

- Till now, We saw 2 Methods to send a signal
  - Using Keyboard as “Ctrl+C”
  - Using Terminal Commands “Kill”
- A 3rd method is to send a signal from code
  - A process sends a signal to another process or itself
- Kill(PID, SIG) => sends a signal “SIG” to process with pid=”PID”
- Raise(SIG) => sends a signal to itself

Try Signal04.c

- How many times did he call the handler?

Parent Handlers

- The Handlers defined in the parent process before the fork (before creating the child) is inherited
- Try Signal05.c

killpg

- Send a signal to process group
- What is the process group? And how to get its id?
- killpg(getpgrp(), sig)
- Try Signal06.c

Alarm

- alarm(N) creates a timer that waits for N seconds then send a “SIGALRM” signal
- It is useful to check timeouts
- To disable a timer call alarm(0)
- Try Signal07.c