SIGNALS

ADVANCED PROGRAMMING

SIGNALS

- A software generated interrupt that is sent to a process by the OS to make it aware of an event, such as: when user press Ctrl+C or another process wants to send a message to this process.
- Signals are identified by integers but also have symbolic names.

SIGNALS (STANDARD)

Name	Meaning
SIGABRT	"Abort", abnormal termination
SIGFPE	Floating point exception
SIGILL	"Illegal", invalid instruction
SIGINT	"Interrupt", interactive attention request sent to the program
SIGSEGV	"Segmentation violation", invalid memory accessTrace trap
SIGTERM	"Terminate", termination request sent to the program

SIGNAL STRUCTURES

- The OS keeps two integers (32 bits) to keep up with 32 signals:
  - Pending signals
  - Blocked signals
- Each bit corresponds to a signal number

SIGNAL PROCESSING

- A signal is sent to a process
- The corresponding bit in the pending signals integer for the process is set
- When the OS selects a process to be run on a processor, it checks the pending and blocked integers
  - NO signals pending: the process is restarted normally
  - 1 or more signals pending but blocked: the process is restarted normally
  - 1 or more signals pending and NOT blocked: the process executes code to handle the signals

DEFAULT SIGNAL HANDLERS

- Each signal is associated with one default handler routine
- The default handlers usually:
  - **Ign**: Ignore the signal
  - **Term**: terminate the process
  - **Cont**: unblock a stopped process
  - **Stop**: block the process

USER DEFINED SIGNAL HANDLERS

- The programmer can replace almost all default signal handlers for his/her code
  - **Except: SIGKILL, SIGSTOP**
- A signal handler function can have any name, but must have a void return type and receive one int parameter
- Example:

```
void handle_sigint(int sig) {
    printf(“Signal number: %d\n”, sig);
}
```

FUNCTIONS

Function	Operation
raise	Artificially raises a signal
signal/sigaction	Sets the action taken when the program receives a specific signal
kill	Sends a SIGKILL signal to the process with the specified ID, which should cause it to terminate

REFERENCES

- Kadam Patel. Signals in C language. Retrieved from: https://www.geeksforgeeks.org/signals-c-language/