

## Лабораторная работа № U5

### Сигналы и их обработка

Сигналы используются для взаимодействия между процессами. При этом сигнал можно отправлять как от процесса **А** процессу **Б**, так и пользователем из командной строки процессу **Б**.

Команды (в ОС unix):

kill        отправка сигнала по PID (идентификатору процесса)

killall    отправка сигналов нескольким процессам по их имени

С т.з. программ сигналы представляют собой некоторую очередь, куда помещаются записи о событии, а программа при получении процессорного времени обрабатывает эти события последовательно специальными обработчиками. Причём любая программа, даже т.н. "HelloWorld" может обрабатывать сигналы.

#### 1. Перечень доступных кодов сигналов: kill -l

Частичный список

1. SIGHUP        9. SIGKILL        15. SIGTERM

#### 2. Отправка сигналов kill -HUP pid или kill -1 pid, где pid – идентификатор процесса

kill 1010 и kill -TERM 1010 являются идентичными – завершение работы процесса

или killall -HUP sendmail - отправка сигнала группе процессов(программ) с именем sendmail

#### 3. перехват сигналов в программе-скрипте, команда trap

Набрать скрипт и проверить его работу:

```
#!/bin/bash
trap "echo Signal - SIGINT" INT
trap "echo Signal - SIGTERM" TERM
echo "pid is $$"
while :
do
    sleep 1
done
# This is the same as "while true".
# This script is not really doing anything.
```

#### 4. перехват сигнала в программе на C - конструкция signal

sighandler\_t signal(int signum, sighandler\_t handler), где signum это константа-номер сигнала (4) или его буквенное обозначение (SIGILL), а в sighandler\_t указывается функция-обработчик сигнала.

#### Пример реализации:

```
void sig_handler (int signum)
{
    if (signum==SIGILL)
    {
        printf("SIGILL received\r\n");
        exit(0);      /*завершаем работу программы*/
    }
}

void main()
{
    ...
    /*назначаем функцию-обработчик сигналов*/
    signal(SIGILL, sig_handler);
    signal(..., ....);
    /*основная логика*/
    ....
}
```

}

### Допинформация:

Для signal – нужна библиотека signal.h , для exit – нужна библиотека stdlib.h

Компилировать программу gcc -o outputfile inputfile, где inputfile имя файла с программой на “C”(расширение файла - .c ) , outputfile имя для создаваемого исполняемого файла

### Задание:

- 1) Написать 2 программы.
- 2) Программа 1: Повторить программу из занятия №4 «управление процессами» [программа с вечным циклом] без какой-либо реализации обработки сигналов.
- 3) Программа 2: Доработать программу из занятия №4 «управление процессами» [программа с вечным циклом] для реализации обработки сигналов (варианты по компьютерам):
  - А) SIGHUP вывести сообщение и не завершаться, SIGUSR1 просто завершиться
  - Б) SIGILL завершиться, SIGTERM вывести сообщение
  - В) SIGTERM вывести сообщение и завершиться, SIGUSR2 вывести сообщение
- 4) запуская программу №1 проверить реакцию программы на получение сигналов (SIGHUP, SIGUSR1, SIGILL, SIGTERM, SIGUSR2), заполнить таблицу результатов
- 5) запуская программу №2 проверить реакцию программы на получение сигналов (SIGHUP, SIGUSR1, SIGILL, SIGTERM, SIGUSR2), заполнить таблицу результатов

Сигнал	Программа 1 (без обработки сигналов)	Программа 2 (с обработкой сигналов)
<b>SIGHUP</b>		
<b>SIGUSR1</b>		
<b>SIGILL</b>		
<b>SIGTERM</b> (просто kill проверить)		
<b>SIGUSR2</b>		