МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«Харківський Політехнічний Інститут»

Кафедра Стратегічного управління

ЗВІТ

з лабораторної роботи № 4

з дисципліни «Операційні системи»

на тему «Фонові процеси та сигнали»

Перевірила: старший викладач

Вільхівська О. В.

Виконав: ст. гр. КН-27

Харків, 2018

Л**абораторная работа №4**

**Тема:** Фоновые процессы и сигналы

**Задания**

**1. Разработать приложение для Linux.**

Разработать приложение для Linux, которое  
- запускается как демон (фоновый процесс)  
- отключается от терминала, сообщает о себе в системном журнале и ждет сигналов

демон должен заносить в журнал информацию  
1) о том, что он стартовал,  
2) в случае получения сигнала  
3) при окончании работы по сигналу SIGINT.  
- по сигналу SIGUSR1 записывает информацию о получении сигнала в системный журнал;  
- по сигналу SIGINT завершает выполнение (тоже с записью в журнале)  
Уметь смотреть состояние запущенного демона командой ps и объяснять содержимое  
различных полей вывода ps.

**Ход Работы**

Код программы

|  |
| --- |
| #include <cstdlib>  #include <stdio.h>  #include <unistd.h>  #include <sys/resource.h>  #include <syslog.h>  #include <signal.h>  using namespace std;  void sigCatch (int signum) {  openlog ("lab4", LOG\_PID, LOG\_LOCAL1);  switch (signum) {  case SIGUSR1:  syslog (LOG\_NOTICE, "Catch! - I got signal %d\n", signum);  closelog();  break;  case SIGINT:  syslog (LOG\_NOTICE, "Process ended %d\n", signum);  closelog();  exit(-1);  break;  }  }  int main(int argc, char\*\* argv) {  pid\_t pid;  if ((pid = fork()) < 0) return -1;  if (pid == 0){ //потомок  (void)signal(SIGINT, sigCatch);  (void)signal(SIGUSR1, sigCatch);  setsid();  chdir("/");  struct rlimit flim;  getrlimit(RLIMIT\_NOFILE, &flim);  for (int fd = 0; fd < flim.rlim\_max; fd++) close(fd);  for (; ;) pause();  }  else{// предок  printf ("daemon started with pid=%d\n", pid);  openlog ("lab4", LOG\_PID, LOG\_LOCAL1);  syslog (LOG\_INFO, "daemon started with pid=%d", pid);  syslog (LOG\_NOTICE, "Program started by User %d", getuid ());  closelog();  exit(0);  }  return 0;  } |

Пример работы

Терминал1:

|  |
| --- |
| reacher@reacher-X550CC:~$ tail -f /var/log/syslog  Dec 6 17:00:42 reacher-X550CC lab4[2359]: daemon started with pid=2361  Dec 6 17:00:42 reacher-X550CC lab4[2359]: Program started by User 1000 |

Терминал2:

|  |
| --- |
| kill -usr1 2361 |

Терминал1:

|  |
| --- |
| Dec 6 17:01:52 reacher-X550CC lab4[2361]: Catch! - I got signal 10 |

Терминал2:

|  |
| --- |
| kill -int 2361 |

Терминал1:

|  |
| --- |
| Dec 6 17:03:25 reacher-X550CC lab4[2361]: Process ended 2 |

**Выводы**

При выполнении лабораторной работы я ознакомился с процессами-демонами - фоновыми неинтерактивынми программами, обеспечивающими работу различных сервисов в системе. Одним из важнейших процессов демонов в Linux является init. В ходе выполнения был создан процесс демон. Для этого понадобилось создать копию процесса (операция fork), моментально завершить процесс предок и создать новую сессию вызвав setsid() в потомке. Также был добавлен обработчик сигналов реагирующий на два типа сигналов SIGUSR1 и SIGINT.