

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
В.Н.КАРАЗІНА
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ ТА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ

ЛАБОРАТОРНА РОБОТА № 9
З ДИСЦИПЛІНИ «ОПЕРАЦІЙНІ СИСТЕМИ»

ТЕМА «ПРОЦЕСИ. СИГНАЛИ»

Виконав студент 3курсу, групи
КС31

спеціальності

122 – Комп'ютерні науки

Касьяненко Микита Михайлович

Прийняв:

доцен кафедри шт. ін.

і прогр. забезп. к.н.т.

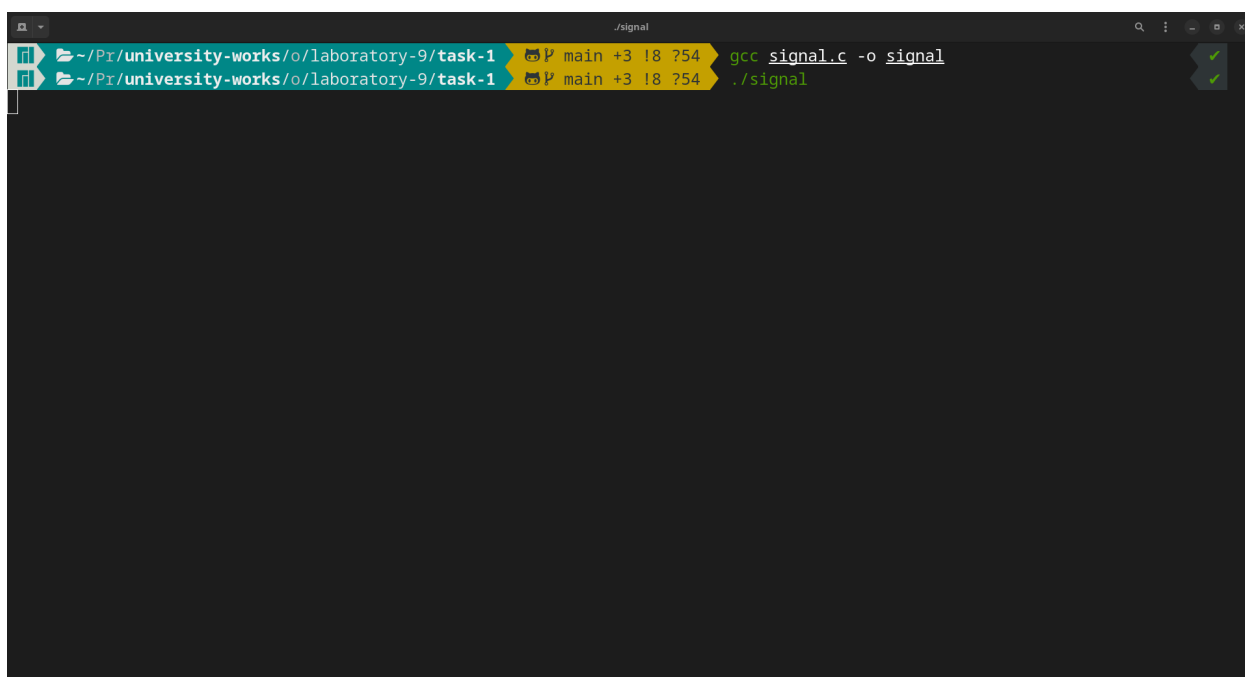
О.Є. Споров ____

Харків 2022

Завдання 1

Напишіть програму, в якій визначте реакцію користувача на сигнали SIGINT, SIGTERM (реакція — виведення інформації про захоплення відповідного сигналу) і сигнал SIGUSR1 (реакція — завершення роботи програми), відновіть стандартну поведінку для сигналу SIGPROF і ігноруйте сигнал SIGHUP. Для затримки виконання процесу використовуйте функцію pause() у нескінченному циклі. Переконайтесь у працездатності програми (за допомогою команди kill командної оболонки надішліть процесу потрібний сигнал і відстежте його реакцією). Реалізуйте дві версії програми: встановлення диспозиції за допомогою функції signal та за допомогою функції sigaction.

Відповідь



```
~/Pr/university-works/o/laboratory-9/task-1 P main +3 18 754 gcc signal.c -o signal
~/Pr/university-works/o/laboratory-9/task-1 P main +3 18 754 ./signal
```

```

mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1

root      23521  0.0  0.0    0    0 ?    I    22:31  0:00 [kworker/0:1-events]
root      23522  0.0  0.0    0    0 ?    I    22:31  0:00 [kworker/4:3]
root      23523  0.0  0.0    0    0 ?    I    22:31  0:00 [kworker/3:2-mm_percpu_wq]
root      23841  0.0  0.0    0    0 ?    I    22:34  0:00 [kworker/u16:0]
mykyta    23892  6.1  1.1 742676 88492 ?    Sl   22:35  0:04 /usr/bin/nautilus --gapplication-service
root      23920  0.0  0.0    0    0 ?    I    22:35  0:00 [kworker/6:2-events]
root      23930  0.0  0.0    0    0 ?    I    22:35  0:00 [kworker/7:1-events]
root      23931  0.0  0.0    0    0 ?    I    22:35  0:00 [kworker/1:2-events]
root      23935  0.0  0.0    0    0 ?    I    22:35  0:00 [kworker/2:1-events]
root      23971  0.0  0.0    0    0 ?    I<   22:35  0:00 [kworker/u17:3-rtw_tx_wq]
mykyta    23995  4.1  0.7 640672 61784 ?    Rsl  22:35  0:00 /usr/lib/gnome-terminal-server
mykyta    24017  2.6  0.1 12840  9124 pts/0  Ss   22:35  0:00 zsh
mykyta    24022  0.0  0.0 10516  4344 pts/0  S    22:35  0:00 zsh
mykyta    24062  0.0  0.0 12116  5204 pts/0  S    22:35  0:00 zsh
mykyta    24063  0.0  0.0 12100  3796 pts/0  S    22:35  0:00 zsh
mykyta    24065  0.0  0.0 858820 4216 pts/0  Sl   22:35  0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
mykyta    24148  0.0  0.0  2356   872 pts/0  S+   22:36  0:00 ./signal
mykyta    24150  7.3  0.1 12836  9060 pts/1  Ss   22:36  0:00 zsh
mykyta    24155  0.0  0.0 10516  4380 pts/1  S    22:36  0:00 zsh
mykyta    24195  0.0  0.0 12116  5268 pts/1  S    22:36  0:00 zsh
mykyta    24196  0.0  0.0 12100  3796 pts/1  S    22:36  0:00 zsh
mykyta    24198  0.1  0.0 662212 4248 pts/1  Sl   22:36  0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
mykyta    24218  2.5  0.3 468576 30860 ?    SNsl 22:36  0:00 /usr/lib/tracker-extract-3
mykyta    24234  0.0  0.0 10028  3480 pts/1  R+   22:36  0:00 ps aux

~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754  kill 24148
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754

```

```

mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1

~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754  gcc signal.c -o signal
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754  ./signal
Taken SIGTERM!
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754
24s

```

```

~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754 gcc signalaction.c -o signalaction
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754 ./signalaction

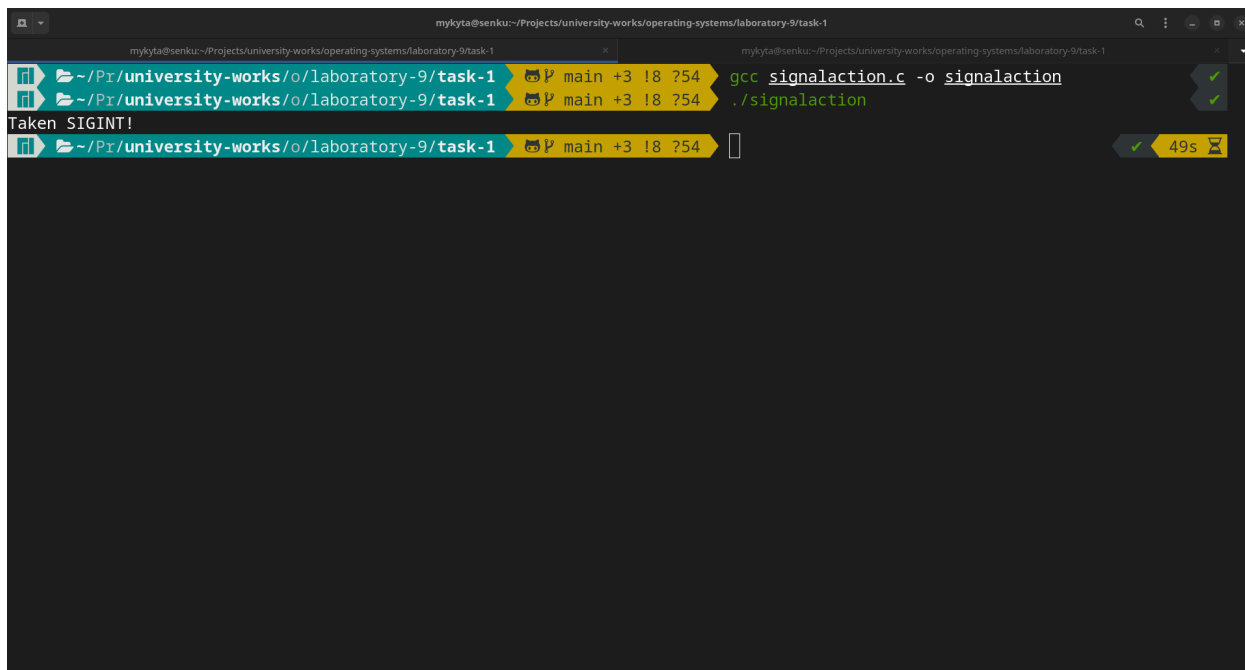
```

```

mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1

mykyta 23995 1.3 0.7 641532 62380 ? Rsl 22:35 0:02 /usr/lib/gnome-terminal-server
mykyta 24017 0.4 0.1 13368 9660 pts/0 Ss+ 22:35 0:00 zsh
mykyta 24022 0.0 0.0 10516 4344 pts/0 S 22:35 0:00 zsh
mykyta 24062 0.0 0.0 12116 5204 pts/0 S 22:35 0:00 zsh
mykyta 24063 0.0 0.0 12100 3796 pts/0 S 22:35 0:00 zsh
mykyta 24065 0.0 0.0 1055428 4216 pts/0 Sl 22:35 0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
mykyta 24150 0.4 0.1 12836 9060 pts/1 Ss+ 22:36 0:00 zsh
mykyta 24155 0.0 0.0 10516 4380 pts/1 S 22:36 0:00 zsh
mykyta 24195 0.0 0.0 12116 5268 pts/1 S 22:36 0:00 zsh
mykyta 24196 0.0 0.0 12100 3796 pts/1 S 22:36 0:00 zsh
mykyta 24198 0.0 0.0 989892 4248 pts/1 Sl 22:36 0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
root 24272 0.0 0.0 0 0 ? I 22:36 0:00 [kworker/5:3-events]
mykyta 24299 1.6 0.1 13840 10232 pts/2 Ss 22:37 0:00 zsh
mykyta 24305 0.0 0.0 10516 4356 pts/2 S 22:37 0:00 zsh
mykyta 24345 0.0 0.0 12116 5184 pts/2 S 22:37 0:00 zsh
mykyta 24346 0.0 0.0 12100 3796 pts/2 S 22:37 0:00 zsh
mykyta 24348 0.0 0.0 858820 4148 pts/2 Sl 22:37 0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
mykyta 24408 0.0 0.0 2356 868 pts/2 S+ 22:38 0:00 ./signalaction
mykyta 24409 6.5 0.1 12836 9196 pts/4 Ss 22:38 0:00 zsh
mykyta 24415 0.0 0.0 10516 4344 pts/4 S 22:38 0:00 zsh
mykyta 24455 0.0 0.0 12120 5208 pts/4 S 22:38 0:00 zsh
mykyta 24456 0.0 0.0 12104 3800 pts/4 S 22:38 0:00 zsh
mykyta 24458 0.0 0.0 727748 4200 pts/4 Sl 22:38 0:00 /usr/share/zsh-theme-powerlevel10k/gitstatus/usrbin/gitstat
mykyta 24486 0.0 0.0 10028 3480 pts/4 R+ 22:38 0:00 ps aux
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754 kill -s SIGINT 24408
~/Pr/university-works/o/laboratory-9/task-1  P main +3 !8 754

```



```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-1
~/Pr/university-works/o/laboratory-9/task-1 P main +3 !8 754 gcc signalaction.c -o signalaction ✓
~/Pr/university-works/o/laboratory-9/task-1 P main +3 !8 754 ./signalaction ✓
Taken SIGINT!
~/Pr/university-works/o/laboratory-9/task-1 P main +3 !8 754 [ ] ✓ 49s
```

Завдання 2

Напишіть програму «будильник»: програма через параметри командного рядка отримує потрібний інтервал часу (в секундах, для зручності налагодження та демонстрації) та текстове повідомлення; завершує основний процес, а в дочірньому за допомогою функцій `alarm` та `pause` через заданий час виводить у стандартний потік виведення заданий текст.

Відповідь

```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-2
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754 gcc main.c -o main
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754 ./main 10 Hello
Main Thread Exit.
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754
```

```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-2
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754 gcc main.c -o main
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754 ./main 10 Hello
Main Thread Exit.
~/Pr/university-works/o/laboratory-9/task-2 main +3 !8 754 Hello
```

Завдання 3

Напишіть програму, в якій створюються два процеси (батьківський та дочірній). Ці процеси мають по чергово, синхронізовано виводити повідомлення в стандартний потік виведення. Процеси синхронізують свою роботу, надсилаючи один одному сигнал `SIGUSR1` за допомогою виклику `kill` (замість сигналу `SIGUSR1` можна взяти один із сигналів реального часу).

Відповідь

```

mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-3
~/Pr/university-works/o/laboratory-9/task-3 P main +3 !8 754 gcc main.c -o main
~/Pr/university-works/o/laboratory-9/task-3 P main +3 !8 754 ./main
0. Taken SIGUSR1!
Parent process:
0. Taken SIGUSR1!
Child process:
1. Taken SIGUSR1!
Parent process:
1. Taken SIGUSR1!
Child process:
2. Taken SIGUSR1!
Parent process:
2. Taken SIGUSR1!
Child process:
3. Taken SIGUSR1!
Parent process:
3. Taken SIGUSR1!
Child process:
4. Taken SIGUSR1!
Parent process:
4. Taken SIGUSR1!
Child process:
^C
~/Pr/university-works/o/laboratory-9/task-3 P main +3 !8 754 [INT X 4s]

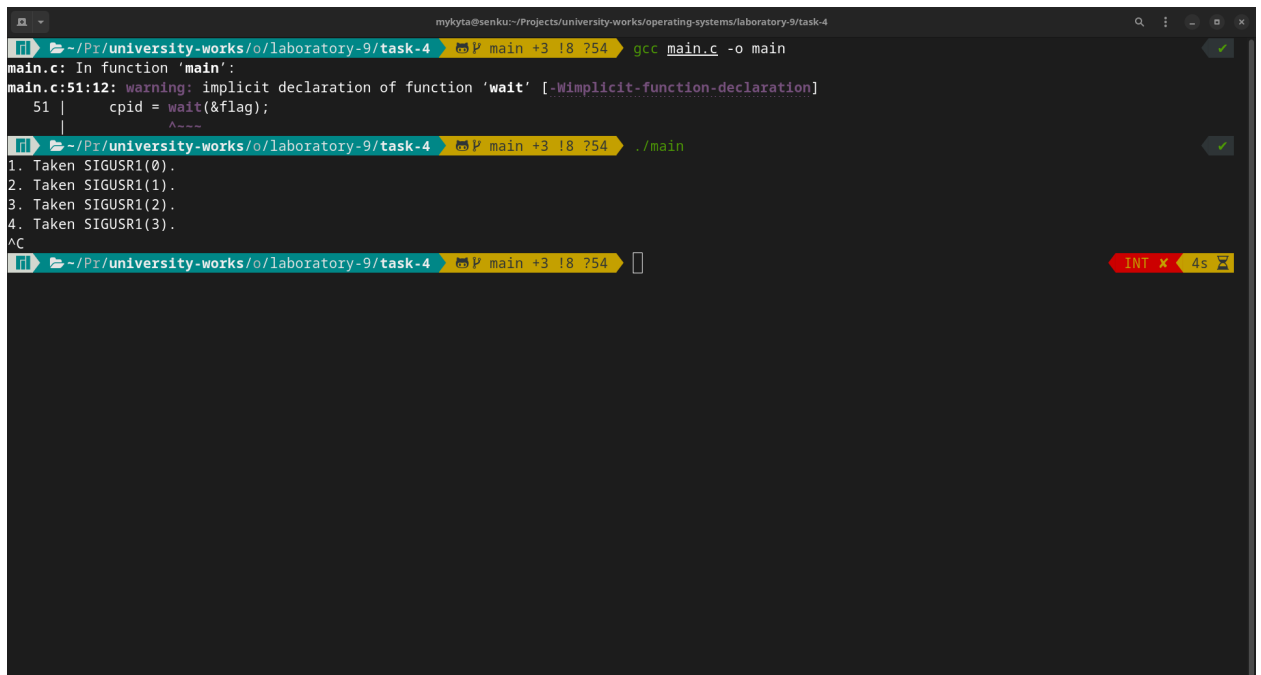
```

Завдання 4

Напишіть програму, в якій створюються два процеси (батьківський та дочірній). Батьківський процес задану кількість разів надсилає дочірньому процесу один із сигналів реального часу з додатковою інформацією (ця інформація — ціле число - номер виклику). Дочірній процес обробляє сигнал і стандартний потік виведення виводить

отриманий номер, текстове подання сигналу та додаткову інформацію. Потім основний процес посилає дочірньому сигнал SIGTERM, що завершує його роботу, чекає на завершення дочірнього процесу і завершує роботу сам.

Відповідь



```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-9/task-4
~/Pr/university-works/o/laboratory-9/task-4 main +3 18 754 gcc main.c -o main
main.c: In function 'main':
main.c:51:12: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
   51 |     cpid = wait(&flag);
       |           ^~~~
~/Pr/university-works/o/laboratory-9/task-4 main +3 18 754 ./main
1. Taken SIGUSR1(0).
2. Taken SIGUSR1(1).
3. Taken SIGUSR1(2).
4. Taken SIGUSR1(3).
^C
~/Pr/university-works/o/laboratory-9/task-4 main +3 18 754 [INT x 4s]
```