

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №3

Выполнил студент гр. 150502:

Бейнар А.В.

Проверил:

Поденок Л.П.

Минск 2023

Тема работы: Взаимодействие и синхронизация процессов.

Цель работы: Синхронизация процессов с помощью сигналов и обработка сигналов таймера. Управление дочерними процессами и упорядочение вывода в stdout от них, используя сигналы SIGUSR1 и SIGUSR2.

Задание:

Действия родительского процесса

По нажатию клавиши «+» родительский процесс (P) порождает дочерний процесс (C_k) и сообщает об этом.

По нажатию клавиши «-» P удаляет последний порожденный C_k, сообщает об этом и о количестве оставшихся.

При вводе символа «k» P удаляет все C_k и сообщает об этом.

При вводе символа «s» P запрещает всем C_k выводить статистику (см. ниже).

При вводе символа «g» P разрешает всем C_k выводить статистику.

При вводе символов «s<num>» P запрещает C_<num> выводить статистику.

При вводе символов «g<num>» P разрешает C_<num> выводить статистику.

При вводе символов «p<num>» P запрещает всем C_k вывод и запрашивает C_<num> вывести свою статистику. По истечению заданного времени (5 с, например), если не введен символ «g», разрешает всем C_k снова выводить статистику.

По нажатию клавиши «q» P удаляет все C_k, сообщает об этом и завершается.

Действия дочернего процесса

Дочерний процесс во внешнем цикле заводит будильник (nanosleep(2)) и входит в вечный цикл, в котором в режиме чередования заполняет структуру, содержащую пару переменных типа int, значениями {0, 0} и {1, 1} (см. раздел «Проблемы неатомарного доступа» темы «Сигналы»).

При получении сигнала от будильника проверяет содержимое структуры, собирает статистику и повторяет тело внешнего цикла.

Через заданное количество повторений внешнего цикла (например, через 101) дочерний процесс, если ему разрешено, выводит свои PPID, PID и 4 числа — количество разных пар, зарегистрированных в момент получения сигнала от будильника.

Вывод осуществляется посимвольно.

C_k запрашивает доступ к stdout у P и осуществляет вывод после подтверждения. По завершению вывода C_k сообщает P об этом.

Следует подобрать интервал времени ожидания и количество повторений внешнего цикла, чтобы статистика была значимой.

Вывод в консоли после компиляции

```
[+] - create child process
[-] - kill last child process
[k] - kill all add child processes
[s]<num> - stop statistic
[g]<num> - resume statistic
[p]num - print C_{num} statistic and freeze other childs
[q] - exit
```

```
+
Child with pid=36221 was created
ppid=36202, pid=36221, stat={22, 20, 22, 37}
36221 informed parent
ppid=36202, pid=36221, stat={29, 29, 20, 23}
ppid=36202, pid=36221, stat={24, 21, 29, 27}
ppid=36202, pid=36221, stat={29, 20, 27, 25}
```

```
-
Process with pid=36221 is ended
Last child was killed, 0 left
```

```
k
Process with pid=36556 is ended
Process with pid=36555 is ended
All child process was killed
```

```
s0
ppid=36202, pid=36906, stat={27, 17, 27, 30}
ppid=36202, pid=36905, stat={24, 28, 29, 20}
36905 informed parent
ppid=36202, pid=36906, stat={18, 32, 25, 26}
ppid=36202, pid=36906, stat={18, 27, 23, 33}
ppid=36202, pid=36906, stat={27, 22, 21, 31}
```

```
g0
ppid=36202, pid=36906, stat={21, 31, 25, 24}
ppid=36202, pid=36905, stat={27, 22, 29, 23}
ppid=36202, pid=36906, stat={26, 23, 27, 25}
ppid=36202, pid=36905, stat={18, 26, 31, 26}
```

```
p0
ppid=36202, pid=37416, stat={20, 28, 24, 29}
ppid=36202, pid=36906, stat={26, 24, 13, 38}
ppid=36202, pid=36905, stat={24, 30, 22, 25}
```

```
q
Process with pid=37416 is ended
Process with pid=36906 is ended
Process with pid=36905 is ended
Kill all child process complete! Finish!
```