



12:02:33 / RUNNING

Submit a solution for mz14-1-mz14-1

Full score:	100
Run penalty:	10
Time limit:	400 ms
Real time limit:	2 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-11-28 10:30:00
Deadline:	2022-11-28 12:05:00

Problem mz14-1: mz14-1

Программа должна напечатать на стандартный поток вывода свой PID, после чего перейти в режим ожидания сигналов. Программа должна обрабатывать сигнал SIGHUP. На каждый приход сигнала программа должна напечатать номер поступления сигнала (0, 1, 2) – каждый раз на отдельной строке. На шестой приход сигнала программа не должна ничего выводить, а просто завершиться с кодом завершения 0.

Таким образом, программа всегда должна выводить:

```
СВОЙ-PID
0
1
2
3
4
```

Стандартный ввод и стандартный вывод программы будут перенаправлены. Не забывайте выводить разделитель и сбрасывать буфер вывода. С другой стороны каналов находится другой процесс, который начнет свою работу как только получит pid процесса.

Вывод на стандартный поток вывода разместите в обработчике сигнала.

Вариант семантики сигналов (SysV или BSD), используемый функцией signal, неизвестен.

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File Файл не выбран

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
4672	2022-11-28 11:02:28	456	mz14-1	gcc-32	Pending review	2	70=100-3*10	View	View
4665	2022-11-28 11:00:26	458	mz14-1	gcc-32	Partial solution	0	0=0-2*10	View	View
4653	2022-11-28 10:57:50	456	mz14-1	gcc-32	Partial solution	0	0=0-1*10	View	View
4643	2022-11-28 10:52:15	414	mz14-1	gcc-32	Partial solution	0	0	View	View
4642	2022-11-28 10:51:44	429	mz14-1	gcc-32	Coding violation style	N/A	N/A	View	View

View [all](#) / [necessary](#).

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

ht10-2

ht10-4

up13-4

up13-5

mz14-1

mz14-2

mz14-3

mz14-4

mz14-5



12:02:50 / RUNNING

Submit a solution for mz14-2-mz14-2

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	2 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-11-28 10:30:00
Deadline:	2022-11-28 12:05:00

Problem mz14-2: mz14-2

Программа должна напечатать на стандартный поток вывода свой pid.

Затем программе на стандартном потоке ввода подаются 32-битные знаковые целые числа. В зависимости от текущего режима работы введенное число либо прибавляется к текущему накопленному значению, либо умножается на текущее накопленное значение. После этого новое накопленное значение выводится на стандартный поток вывода. Переполнение при выполнении 32-битных операций должно игнорироваться, то есть сохранять младшие 32 бита результата, но не приводить к UB. Программа начинает работу в режиме сложения. Начальное накопленное значение равно 0.

При получении сигнала SIGINT программа переключается в режим сложения. При получении сигнала SIGQUIT программа переключается в режим умножения.

Стандартный ввод и стандартный вывод программы будут перенаправлены. Не забывайте сбрасывать буфер вывода. С другой стороны каналов находится другой процесс, который начнет свою работу как только получит pid процесса. Можете рассчитывать, что сигналы обрабатываются по модели BSD.

Пример дан в предположении, что программе не отправляются сигналы.

Examples

Input

100 200 -44 500

Output

100
300
256
756

Submit a solution

Language:gcc-32 - GNU C (32 bit) 12.1.1

File Файл не выбран

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
4713	2022-11-28 11:26:01	981	mz14-2	gcc-32	Pending review	7	100	View	View
4712	2022-11-28 11:25:36	1067	mz14-2	gcc-32	Coding violation style	N/A	N/A	View	View

View
[all](#) / [necessary](#)

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

ht10-2

ht10-4

up13-4

up13-5

mz14-1

mz14-2

mz14-3

mz14-4

mz14-5



12:03:08 / RUNNING

Submit a solution for mz14-3-mz14-3

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-11-28 10:30:00
Deadline:	2022-11-28 12:05:00

Problem mz14-3: mz14-3

В аргументах командной строки процессу передается число N и имена текстовых файлов. Каждый текстовый файл должен содержать одну строку — имя исполняемого файла. Процесс запускает на параллельное исполнение не более чем первые N исполняемых файлов и после окончания параллельного исполнения на последовательное исполнение — оставшиеся исполняемые файлы. По окончании работы процесс-родитель выводит на экран число — количество потомков, которые были успешно запущены и завершили свою работу с пользовательским кодом 0.

Единственная строка каждого текстового файла содержит символ '\n' в конце. В строке отсутствуют пробельные символы, кроме финального '\n'.

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File файл не выбран

Send!

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
4774	2022-11-28 12:00:40	1234	mz14-3	gcc-32	Pending review	7	90=100-1*10	View	View
4765	2022-11-28 11:58:39	1166	mz14-3	gcc-32	Partial solution	1	0	View	View

View all / necessary
up02-4
up02-5
up02-6
up02-7
mz09-1
mz09-2
mz09-3
mz09-4
mz09-5
mz09-6
ht10-2
ht10-4
up13-4
up13-5
mz14-1
mz14-2
mz14-3
mz14-4
mz14-5



12:01:54 / RUNNING

Submit a solution for mz14-4-mz14-4

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-11-28 10:30:00
Deadline:	2022-11-28 12:05:00

Problem mz14-4: mz14-4

Родитель создает двух сыновей, связывая их каналом. Сыновья начинают обмениваться числами 1, 2, 3, ... Число 1 получает первый сын. При получении числа из входного канала каждый сын печатает на стандартный поток вывода родителя свой номер (1 или 2) и полученное число и пересылает обратно число, на 1 большее. При достижении числа, задаваемого в командной строке, пересылает то же самое число в канал и завершает работу.

Процесс должен завершить работу при получении максимального числа из канала. Завершать процесс в других случаях не разрешается. Запрещается отправлять сигнал отцу.

Отец дожидается завершения работы обоих процессов, затем выводит строку Done и завершает работу.

Если в командной строке передано число 5, pid первого процесса - 100, pid второго процесса - 101, то вывод должен быть таким:

```
1 1
2 2
1 3
2 4
Done
```

Данные передавать в текстовом виде, для ввода-вывода использовать высокоуровневый ввод-вывод (printf, scanf).

Для синхронизации использовать единственный сигнал SIGUSR1. PID процессов передавать через канал.

Привязывать файловые дескрипторы к дескрипторам потока можно либо с помощью функции fdopen, либо с помощью перенаправления стандартных потоков ввода-вывода.

Какие-либо средства взаимодействия, кроме одного канала и сигнала SIGUSR1 использовать запрещено. signalfd использовать запрещено.

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File файл не выбран

Send!

View
[all](#) / [necessary](#)

up02-4

up02-5

up02-6

up02-7

mz09-1

mz09-2

mz09-3

mz09-4

mz09-5

mz09-6

ht10-2

ht10-4

up13-4

up13-5

mz14-1

mz14-2

mz14-3

mz14-4

mz14-5



12:02:20 / RUNNING

Submit a solution for mz14-5-mz14-5

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	2 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022-11-28 10:30:00
Deadline:	2022-11-28 12:05:00

Problem mz14-5: mz14-5

Программе на стандартном потоке ввода задаются два целых числа low и high. Программа должна вывести на стандартный поток вывода свой PID, после вывода программа должна обрабатывать получение сигналов SIGINT и SIGTERM.

Программа должна вычислять простые числа в интервале [low,high). Если программа получила сигнал SIGINT, программа должна вывести на стандартный поток вывода последнее на текущей момент найденное простое число. При получении SIGINT в четвертый раз, программа должна закончить работу с кодом завершения 0. Таким образом, может быть выведено не более трех простых чисел.

При получении сигнала SIGTERM программа должна завершиться с кодом 0.

Если программа закончила вычисления, программа должна вывести на стандартный поток вывода число -1 и завершиться с кодом 0.

Если на момент прихода сигнала ни одно простое число найдено не было, выводите 0.

Учтите, что стандартный поток вывода программы будет перенаправлен в pipe.

Submit a solution

Language: gcc-32 - GNU C (32 bit) 12.1.1

File Файл не выбран

Send!

View
[all](#) / [necessary](#)

up02-4
up02-5
up02-6
up02-7

mz09-1
mz09-2

mz09-3
mz09-4
mz09-5
mz09-6

ht10-2
ht10-4
up13-4
up13-5

mz14-1
mz14-2
mz14-3

mz14-4
mz14-5