

How can a child process return two values to the parent when using pipe()?

Love Stack Overflow? Love building stuff?

Use our API to build your own mobile, web, and desktop Q&A apps
— your platform, your rules!

Hello World!

This is a collaboratively edited question and answer site for **professional and enthusiast programmers**. It's 100% free, no registration required.

[about »](#) [faq »](#)



I have my child process counting the frequency of words from a text file. I am using `pipe()` for IPC. How can the child process return both the word name and the word frequency to the parent process? My source code is in C and I am executing it in a UNIX environment.

[c](#) [unix](#) [ipc](#) [pipe](#)

[link](#) | [improve this question](#)

edited 11 mins ago
 Jonathan Grynspan
12k • 1 • 8 • 26

asked 19 mins ago
 het
11 • 2

1 What have you tried so far? Do you have any code that you cannot get to work? – [Vlad Lazarenko](#) 17 mins ago

Do you know how to create the pipe? Do you know how to send anything? – [Beta](#) 9 mins ago

[feedback](#)

tagged

[c](#) × 40918

[unix](#) × 6069

[ipc](#) × 773

[pipe](#) × 439

asked today

viewed 18 times

active today

2 Answers

[active](#)

[oldest](#)

[votes](#)



Writes to a pipe up to the size of `PIPE_BUF` are atomic (included in `limits.h`), therefore you can easily pack your information into some type of struct, and write that to the pipe in your child process for the parent process to read. For instance, you could setup your struct to look like:

```
struct message
{
    int word_freq;
    char word[256];
};
```

Then simply do a read from your pipe with a buffer that is equal to `sizeof(struct message)`. That being said, keep in mind that it is best to only have either a single reader/writer to the pipe, or you can have multiple writers (because writes are atomic), but again, only a single reader. While multiple readers can be managed with pipes, the fact that reads are *not* atomic means that you could end up with scenarios where messages either get missed due to the non-deterministic nature of process scheduling, or you get garbled messages because a process doesn't complete a read and leaves part of a message in the pipe.



CAREERS 2.0

Join the Expedition! Explore Front-End UX at the Bleeding...
Occipital
Boulder, CO; San Francisco, CA