

# Содержание

- IPC ..... 3
  - signals** ..... 3
    - pid ..... 3
    - socket ..... 3
  - named pipe** ..... 3



# IPC

IPC - **inter-process communications** - межпроцессное взаимодействие

## signals

Сигналы - часть [стандарта POSIX](#), удобны для межпроцессного взаимодействия в unix-like системах.

Их может посылать как пользователь программе напрямую<sup>1)</sup>, так и *master* процесс: когда основная программа - демон, называемый *worker*, а сигналы ему шлет *master*, как, например, у *ntpd* и *ntpd*.

Мастер должен найти процесс-воркер, что бы слать ему сигналы.

## pid

PID (**process identifier**) - уникальный числовой идентификатор процесса. Чтобы *master*'у получить PID *worker*'а: - Использовать команды, такие как *ps* или *pgrep*. Например, команда *ps aux* выводит список всех процессов с их PID, что позволяет мастеру определить, какой процесс-воркер необходимо адресовать. Но для этого надо взаимодействовать с shell'ом. Надежнее, но заморочнее. - Прочитать его из файла<sup>2)</sup> в условленном месте, куда *worker* при старте записывает свой PID. Если файла в условленном месте нет - *master* считает, что *worker* не запущен.

 **Fix Me!** примеры

## socket

Сокеты - еще один способ межпроцессного взаимодействия. Сокеты могут быть как «локальными», использующими файловую систему, так и «сетевыми», использующими сетевые протоколы. Сокеты могут использовать различные протоколы<sup>3)</sup>.

 **Fix Me!** примеры

## named pipe

 **Fix Me!**

1)

вернее ручками, тк дефакто, посылает их *kill*, тоже программа

2)

традиционно, такие файлы имеют суффикс .pid

3)

те же TCP / UDP

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

[https://wiki.radi0.cc/glossary:prog:interprocessing\\_interaction](https://wiki.radi0.cc/glossary:prog:interprocessing_interaction)

Last update: **2025/11/09 12:07**

