

# АЛГОРИТМ РАБОТЫ ПРОГРАММЫ БЫСТРОЙ ДИАГНОСТИКИ СИСТЕМЫ

## Харитонова Т.С.

*Харитонова Татьяна Сергеевна – студент,  
кафедра информационных систем и технологий,  
Северный арктический федеральный университет, г. Архангельск*

**Аннотация:** с развитием большого количества прикладного программного обеспечения, пользователи чаще сталкиваются с проблемой «зависания» программ, для диагностики системы используются диспетчеры процессов, позволяющие отследить проблемные процессы. Требуется понять, каким образом эти программы работают.

**Ключевые слова:** процесс, диспетчер процессов, диагностика системы.

В современном мире компьютеры есть практически у каждого человека, мы используем их и на работе, и дома. Зачастую пользователь компьютера сталкивается с проблемой вирусов, «зависания» программ, шумом компьютера, причина может крыться в процессах, запущенных на данном компьютере. Процесс – программа, которая выполняется в текущий момент. Именно для решения подобных проблем, разрабатываются программы, позволяющие провести быструю диагностику системы. В современных операционных системах обязательно есть стандартная утилита, отвечающая за такую диагностику. Подобные программы называют диспетчерами процессов. Более опытные пользователи компьютеров обращаются к ней довольно часто, однако принцип работы известен далеко не всем.

Рассмотрим подробнее работу диспетчера процессов на базе Linux, составим алгоритм работы, на базе которого можно разработать подобную утилиту.

Для начала, проанализируем предметную область, рассмотрим существующие утилиты – диспетчеры процессов. Так, на базе Linux существует несколько подобных программ: Ps, Top.

Ps - программа в UNIX, Unix-подобных и других POSIX-совместимых операционных системах, выводящая отчёт о работающих процессах, она делает «фото», работающих в данную секунду процессов, кроме того, вывод того что нам именно нужно можно настроить, с помощью вызова из консоли, так, например, запрос «ps -e» выдаст информацию о PID – идентификатор процесса, TTY – терминал с которым связан данный процесс, TIME – процессорное время занятое этим процессором, CMD – команда запустившая данный процесс. Возможны и другие варианты запуска.

Альтернативой PS является утилита top, которая выводит на экран постоянно обновляющуюся информацию, однако ограниченную размерами окна, что не может поместиться на экран – игнорируется. Так же из консоли запустим программу top, которая выдаст информацию о PID – идентификаторе процесса, USER – пользователе, от которого запущен процесс, PR – текущем приоритете процесса, NI – приоритете, выставленным командой nice (от -20 (наивысший) до 19), VIRT – полном объеме виртуальной памяти, которую занимает процесс, RES – текущее использование оперативной памяти, S – текущем состоянии, CPU – проценте доступного времени процессора, MEM – проценте использования оперативной памяти данным процессом, TIME – времени использования процессора в секундах, COMMAND – команде, запустившей процесс.

Утилитам требуется собирать информацию о процессах, каким образом они это делают? На самом деле, вся информация хранится в виртуальном каталоге, так для получения сведений о процессах, состоянии и конфигурации ядра и системы можно узнать из каталога /proc [1]. В этом и есть основной секрет данных программ. Например, состояние памяти можно узнать из файла /proc/meminfo, а состояние процессора из /proc/cpuinfo. Данные же о процессах получить сложнее. На каждый процесс в компьютере в виртуальном каталоге /proc создается отдельный каталог с именем PID процесса – его идентификатором, и в каждом таком каталоге уже содержатся файлы с информацией о процессе. Для получения стандартной информации типа PID, TTY, TIME, CMD следует читать файл status, содержащийся в каталоге /proc/PID.

Таким образом, когда нам надо узнать о работающих процессах, требуется: перебрать существующие в данный момент в системе PID (виртуальные каталоги /proc), в них найти файл status, выбрать из него нужную информацию, правильно вывести ее пользователю. Данную программу проще всего реализовать на языке программирования C в консольной версии под операционную систему Linux.

### Список литературы

- 1 OpenNET. [Электронный ресурс]. Режим доступа: [www.opennet.ru/docs/RUS/bash\\_scripting\\_guide/x12987.html](http://www.opennet.ru/docs/RUS/bash_scripting_guide/x12987.html) (дата обращения: 01.03.2017).