Часть 8.
Отладка приложений

Влад 'mend0za' Шахов
Linux & Embedded Team Leader

Linux & Embedded Department

Лимиты для процесса

ulimit

Встроенная команда Shell, позволяющая контролировать количество ресурсов, выделяемых процессам, запускающимся из Shell.

Лимиты для процесса

ulimit

Встроенная команда Shell, позволяющая контролировать количество ресурсов, выделяемых процессам, запускающимся из Shell.

Что можно контролировать

- время выполнения (cpu time)
- максимальный размер данных (data) и стека (stack)
- количество открытых файлов (open files)
- максимальный размер core dump (core file)

Примечание: для обычного пользователя работают только в сторону ужесточения, без возможности отката.

Лимиты для процесса

ulimit

Встроенная команда Shell, позволяющая контролировать количество ресурсов, выделяемых процессам, запускающимся из Shell.

Что можно контролировать

- время выполнения (cpu time)
- максимальный размер данных (data) и стека (stack)
- количество открытых файлов (open files)
- максимальный размер core dump (core file)

Примечание: для обычного пользователя работают только в сторону ужесточения, без возможности отката.
Упражнение 1: задать лимиты памяти (virtual memory, max memory size), чтобы vim не хватало для запуска
Упражнение 2: задать лимиты времени выполнения

Отладочная информация

Программа может быть скомпилирована с отладочной информацией: именами переменных и функций.

Просмотр отладочной информации

- nm отладочные символы
- ldd список динамических библиотек, используемых данной программой или библиотекой

Часто отладочная информация упаковывается мантейнерами в отдельные пакеты[1]

Упражнение 1 Просмотреть отладочную информацию для vim

Упражнение 2 Просмотреть список динамических библиотек для vim

[1] Суффиксы -dbg, -debug. Например libc6-dbg, vim-dbg

Core dump file

Файл, содержащий образ памяти процесса на момент прерывания выполнения по сигналу[a]

[a] man 7 signal - список сигналов, вызывающий core dump

Упражнение Установить лимит на размер core больший нуля. Запустить vim в фоне (Shell job). Пристрелить vim с помощью сигнала, вызывающего генерацию core dump.

Core dump file

Файл, содержащий образ памяти процесса на момент прерывания выполнения по сигналу[a]

[a] man 7 signal - список сигналов, вызывающий core dump

Упражнение Установить лимит на размер core больший нуля. Запустить vim в фоне (Shell job). Пристрелить vim с помощью сигнала, вызывающего генерацию core dump.

GDB

Позволяет контролировать ход выполнения программы (запуск, остановка, выполнение по шагам) и проверять её состояние.

Для получения осмысленных данных - требует наличия отладочной информации [3] в программе и библиотеках её использующих.

GDB

Позволяет контролировать ход выполнения программы (запуск, остановка, выполнение по шагам) и проверять её состояние.

Для получения осмысленных данных - требует наличия отладочной информации [3] в программе и библиотеках её использующих.

Наиболее существенные (для нас) функции

- установка точек прерывания (break)
- просмотр стека вызовов (bt или backtrace)
- запуск приложений с параметрами (file и run)
- присоединение к запущенным процессам (attach)
- остановка по сигналам и точкам останова
- продолжение выполнения (cont)

Упражнение 1 Загрузить vim вместе с core-дампом из предыдущего упражнения. Получить backtrace.
Упражнение 2. Подсоединиться к своему сессионному shell. С помощью backtrace определить что он делает сейчас.
Упражнение 3. Загрузить ``ls'' в отладчик. Установить точки прерывания на функцию open. Выполнить ``ls -l'' в отладчике. Проанализировать стек вызовов.

Трассировка вызовов функций

Трассировка вызовов

- strace - системных (ядро)
- ltrace - библиотечных (внешние библиотеки)

Трассировка вызовов функций

Трассировка вызовов

- strace - системных (ядро)
- ltrace - библиотечных (внешние библиотеки)

Упражнение 1 запустить vim под strace, с сохранением вывода в файл

Упражнение 2 запустить vim под ltrace, с сохранением вывода в файл

Упражение 3 подсоединить strace к своему текущему shell, в фоне, с сохранением вывода в файл

Упражение 4 подсоединить ltrace к своему текущему shell, в фоне, с сохранением вывода в файл