

Лабораторная работа

Управление памятью

Цели работы:

- изучить основные методы управления памятью в Linux
- построить карту виртуальной памяти процесса
- приобрести практические навыки работы с инструментами мониторинга использования памяти, настройки параметров виртуальной памяти, тестирования работоспособности оперативной памяти

Теоретические сведения

Управление памятью в Linux

Управление памятью осуществляется ядром операционной системы и включает в себя выделение памяти процессам, освобождение памяти по их завершении, управление таблицей страниц, кеширование данных, управление файлом подкачки, обработку ошибок памяти и др.

В Linux реализован механизм виртуальной памяти. Это означает, что процессы не работают с физическими адресами напрямую, у каждого процесса имеется свое закрытое виртуальное адресное пространство, что позволяет им располагать большим объемом памяти, чем физически доступно в системе, а также обеспечивает изоляцию адресных пространств одновременно выполняющихся процессов. Этот механизм основан на организации совместного функционирования оперативной и внешней памяти. Процесс выполняет обращение к памяти, используя виртуальный адрес, а механизм виртуальной памяти определяет соответствующий ему физический адрес или выполняет операцию подкачки в случае, если требуемая страница отсутствует в оперативной памяти. Фактическое отображение виртуальных адресов на физические выполняет модуль управления памятью (MMU, Memory Management Unit).

Наименьшей единицей управления памятью с точки зрения ядра является страница. Размер страницы зависит от архитектуры процессора, но исторически сложилось так, что размер страницы в Linux по умолчанию составляет 4 КБ. Позднее, начиная с версии ядра 2.6.3, была внедрена поддержка огромных страниц (huge pages) размером в 2 МБ. Ядро использует структуру *struct page* для отслеживания состояния каждой страницы физической памяти в системе.

Информацию о состоянии виртуальных страниц процесса можно получить с помощью команды **ptop**, вызванной с указанием pid процесса.

Преобразование виртуальных адресов в физические

Для каждого процесса в Linux создается отдельная таблица страниц, состоящая из дескрипторов (записей) для каждой виртуальной страницы процесса. В дескрипторе содержится информация о номере физической страницы, в которую загружена данная виртуальная страница, права доступа к странице и другие атрибуты. Для уменьшения расхода оперативной памяти

на хранение служебной информации применяются многоуровневые таблицы страниц.

Виртуальный адрес состоит из нескольких частей, соответствующих иерархии таблицы страниц. Количество индексов и битов, отводимых для каждого индекса в структуре виртуального адреса, зависят от архитектуры процессора и реализации ядра Linux.

Микропроцессоры с архитектурой x86-64 поддерживают четыре уровня таблицы страниц. В настоящее время из 64 разрядов виртуального адреса задействованы только младшие 48 бит. Структура виртуального адреса и иерархия таблицы страниц представлены на рисунке 1:

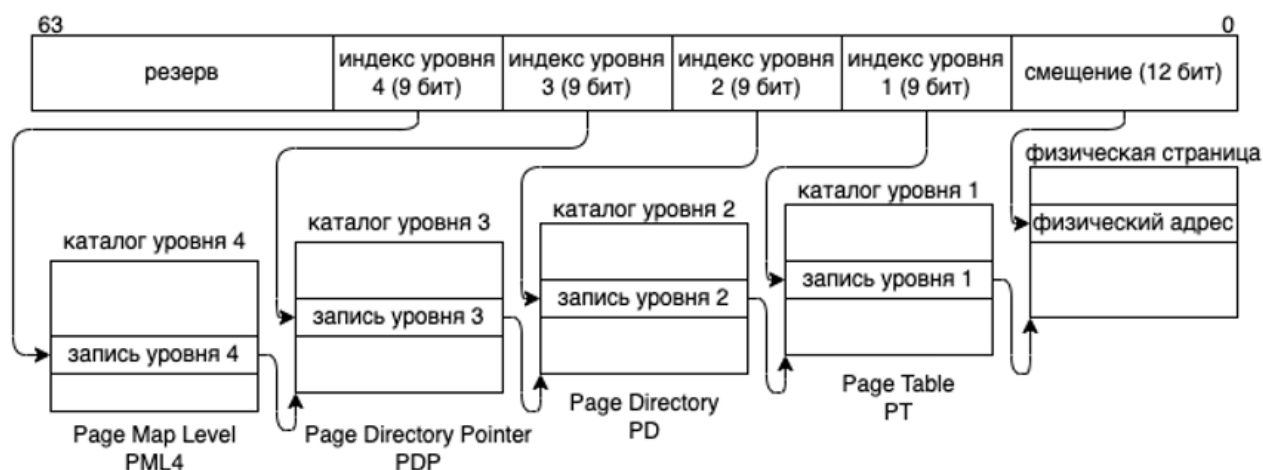


Рис.1. Схема преобразования виртуального адреса в физический в МП x86-64

При трансляции виртуального адреса в физический происходит последовательный переход от уровня к уровню иерархии таблицы страниц, начиная с PML4 и до собственно таблицы страниц, в которой находится дескриптор страницы, соответствующий данному виртуальному адресу. Информация, содержащаяся в этом дескрипторе, используется для преобразования виртуального адреса в физический.

Команды Linux для работы с памятью

top - отображает потребление ресурсов процессами в реальном времени (и др.);

free - утилита для отображения объема свободной и использованной оперативной памяти;

/proc/meminfo — раздел файловой системы /proc содержащий информацию о состоянии памяти в системе:

Пример: **cat /proc/meminfo | grep Mem.**

SWAP

SWAP (или область подкачки) - это часть жесткого диска (отдельный раздел или файл на диске), которая используется операционной системой для хранения временно неиспользуемых фрагментов памяти процессов, перемещенных из оперативной памяти во вторичное хранилище. Дополнительно SWAP используется при организации режима сна (hibernation).

или suspend to disk). В этом случае в SWAP сохраняется образ всей оперативной памяти.

Необходимый объем SWAP определяется в зависимости от объема оперативной памяти вычислительной системы и настроек перехода в спящий режим. Так для систем с ОЗУ до 2 ГБ без перехода в спящий режим под SWAP необходимо выделить 2*ОЗУ, для систем с переходом - 3*ОЗУ; для систем с ОЗУ от 64 ГБ не менее 4 ГБ, причем спящий режим для таких систем не рекомендуется.

Размер раздела подкачки устанавливается автоматически во время установки системы. Для того, чтобы разрешить спящий режим, необходимо отредактировать пространство подкачки на этапе пользовательского разбиения.

Утилиты Linux для работы с SWAP

swapon -s; grep Swap /proc/meminfo; free -h – показывают использование и настройки файла подкачки;

mkswap /swapfile - размечает файл /swapfile как файловую систему swap;

swapon /swapfile - включает/подключает файл подкачки в систему;

cat /etc/fstab - выводит список устройств для монтирования. Можно добавить swap раздел.

Стресс-тестирование памяти

Memtester - утилита, предназначенная для стресс-тестирования памяти компьютера на наличие сбоев. Для корректной работы необходимо запускать утилиту с правами суперпользователя и указывать объем свободной памяти, т.к. при работе утилита перезаписывает память.

Для установки используйте: **apt-get install memtester**.

Запуск проверки свободного блока памяти размера 1024 Мбайт с одним проходом осуществляется командой: **memtester 1024 1**

Во время работы утилиты Memtester имеется возможность отслеживания выходных данных в режиме реального времени в другом окне терминала: **tail -f /var/log/syslog**.

После завершения тестирования данные доступны в системном журнале: **cat /var/log/syslog | grep memtester**.

Порядок выполнения работы

Задание 1. Получение карты виртуальной памяти процесса

С помощью **ptop** выведите карту виртуальной памяти любого работающего процесса, например, экземпляра терминала. Изучите структуру карты, сделайте вывод, какая информация может быть получена в результате работы **ptop**. В отчет включите снимок экрана с полученной картой.

Задание 2. Настройка файла подкачки

1. Проверьте текущее использование памяти с помощью команды **free**.
2. Создайте файл подкачки размером 1 ГБ.

3. Установите требуемые разрешения и отформатируйте файл подкачки с применением **mkswap**.
4. Подключите созданный файл подкачки с применением команды **swapon**.
5. Убедитесь, что размер swar-файла равен 1 ГБ .
6. Добавьте настройку для автоматического подключения swar-файла при перезагрузке виртуальной машины (необходимо внести изменения в файл `/etc/fstab`), перезагрузите машину и убедитесь в корректности работы.
7. Выполните команды **free -h -t** и **swapon -s**.
8. Деактивируйте созданный файл подкачки и удалите его.
9. Проверьте доступное количество свободной памяти.

В отчет включите снимки экрана для каждого пункта задания.

Задание 3. Тестирование памяти

С помощью утилиты Memtester проведите тестирование 1 Гб свободной памяти (если данный объем недоступен, выберите меньшее количество свободной памяти), выполнив не менее двух проходов. Отобразите в отчете процесс и результаты тестирования.

Дополнительное задание

Получите текущее значение параметра **swappiness**. Измените его. Сделайте скриншот внесенного изменения. На что влияет значение этого параметра? В отчете приведите ответ в произвольной форме.

Контрольные вопросы

1. Какие задачи решает подсистема управления памятью?
2. Какой модуль занимается пересчетом виртуальных адресов в физические? Какие данные для работы модуля формирует ОС?
3. Что такое виртуальная память? Каковы преимущества использования ВАП по сравнению с работой с физическими адресами напрямую?
4. Какие адреса отображаются в карте ВАП?
5. Что такое файл подкачки? Как он используется?
6. Какая информация содержится в файле `/etc/fstab`?
7. Для чего используется утилита memtester?
8. Если при выполнении команды **mkswap** в качестве параметра передан файл размером 2Gb с разрешениями 777, что будет выведено в терминал?
9. В чем разница между файлом подкачки и разделом подкачки?
10. Как можно посмотреть количество свободной физической памяти?
Приведите варианты команд.