Московский государственный технический университет им. Н.Э. Баумана

Домашнее задание по дисциплине

**«Операционные системы»**По теме:   
**«Управление виртуальной памятью в ОС Unix»**

**Выполнил**

**студент группы РТ5-51Б \_\_\_\_\_\_\_\_\_\_/Сафиуллин.А.П./**

Оглавление

**Введение**

Unix – семейство переносимых, многозадачных и многопользовательских операционных систем (ОС). История UNIX началась в 1969 году с Кена Томпсона и Денниса Ричи, а также других разработчиков ОС, которые начали разрабатывать данную систему в рамках проекта для подразделения AT&T Bell Laboratories. В течение первых 10 лет развитие UNIX происходило в основном в рамках Bell Labs. Первоначальные версии UNIX (тогда ещё Unics) назывались “Version n” (Vn) и предназначались для 16 битовой ЭВМ DEC PDP-11 и VAX - 32 битовая версия UNIX.

В рамках данной системы была реализована технология виртуальной памяти.

**Концепция виртуальной памяти**

Виртуальная память - стратегия распределения памяти, основанная на идее расширения физической памяти путем размещения расширенной памяти на диске и использования таблиц или сегментов для трансляции адресов. Сама концепция виртуальной памяти основывается на идее отделения логической памяти пользователя от физической памяти и расширение логической памяти путём хранения её образа на диске.

Эта концепция основана на том, что при исполнении программы лишь часть данных, к которым происходит обращение требует размещения в физической памяти. Поэтому, логично, возникает идея расширять пространство логической памяти, которое может быть реализовано намного большего размера, чем имеется в наличии физической памяти. Именно таким образом и реализуется принцип организации виртуальной памяти.

Виртуальная память поддерживает совместное использование одного и того же адресного пространства более чем одним процессом, создание и исполнение облегченных процессов в общем пространстве виртуальной памяти. Виртуальная память обеспечивает более эффективное создание процесса, чем предшествующие ему схемы организации памяти и процессов.

Виртуальная память может быть реализована различными способами:

1. Страничная организация по требованию (paging on demand)
2. Сегментная организация по требованию (segmentation on demand)

Термин «по требованию» подчёркивает динамический характер управления виртуальной памятью, т.е. страница или сегмент будут размещены в ФП только в случае, если к ним происходит обращение из программы. Причём, если размер обрабатываемой области виртуальной память очень велик, то в физической памяти разместится только та страница, к которой обращается пользовательская программа. Логичным, также, будет и то, что виртуальная память должна быть больше физической. Взаимодействие виртуальной и физической памяти происходит при помощи отображения памяти – системной таблицы (сегментов, страниц). Образ виртуальной памяти при этом хранится на диске. В современных реализациях UNIX для большинства архитектур используется страничная организация виртуальной памяти. Принцип реализации памяти в виде страничной организации по требованию заключается в том, что каждая страница загружается в память, только если она требуется при выполнении программы.

Данный подход позволяет уменьшить объем ввода-вывода, т.е. в память подкачивается минимально необходимый объем данных, при этом, очевидно, используется минимально необходимый объем физической памяти. Также, т.к. объем пересылаемых данных мал, то система в среднем быстрее реагирует на каждый подобный запрос. В виду всех этих причин система может обслуживать большое число пользователей.

Основными принципами данной концепции являются следующие:

1. Если страница требуется программе на неё имеется ссылка из программы
2. Если ссылка на страницу не верна, то происходит прерывание
3. Если требуемая страница отсутствует в памяти, то она подкачивается в память. Механизм подкачки происходит через прерывание (page fault).

Всё перечисленное выше позволяет системе при помощи таблицы преобразования адресов поддерживать систему в адекватном, рабочем состоянии. Таким образом, физическая память делится между множеством процессов, при этом каждый из процессов имеет доступ только к «своим» страницам, что гарантируется ОС и аппаратными возможностями процессора, что даёт повышенную безопасность.

В 32-х разрядной архитектуре объем адресного пространства процесса равен 4 гб. Т.к. число процессов в системе гипотетически не ограничено, то суммарный объем адресного пространства может выходить далеко за рамки физической памяти. Однако, большая часть программ оперирует малым объемом памяти. Страничное преобразование позволяет решить проблему недостатка физической памяти: неиспользуемые страницы могут копироваться из физической памяти на диск в специальную область, называемую область подкачки и храниться там до момента, когда они вновь потребуются для исполнения процесса. Также, благодаря такой структуре виртуальной памяти возможен механизм – файлы, проецируемые в память. Т.е. процесс может «присоединить» файл к своему виртуальному адресному пространству. Что означает, что при обращении к файлу или изменении данных ОС будет автоматически выполнять отображение содержимого файла в страницы памяти процесса и обратно.