

Практикум на ЭВМ. Лабораторная работа №2.
8 факультет, 1 курс, 1 семестр 2012/13 учебного года.

Операционная среда ОС UNIX

*«Гремучая змея – дружелюбная особа,
но вы должны вести себя с ней правильно...»
Книга о змеях
(Питер Браун о UNIX)*

*Есть старая поговорка:
«Танцуй с тем, кто тебя привел»,
согласно которой я должен говорить о UNIX.
(Кен Томпсон в момент получения
Тьюринговской премии АСМ).*

Цель работы — изучение и освоение программного обеспечения ОС UNIX и приобретение навыков, необходимых для выполнения курсовых и лабораторных работ в среде UNIX.

Основные этапы работы:

1. Краткое *самостоятельное* **предварительное** изучение литературы по ОС UNIX (например, в объеме первых глав книги Готье [1] или по текстам из интернета/CD-хрестоматии по курсу); желательна также предварительная отработка примеров из книг и данного описания на домашнем компьютере студента.
2. Проверка готовности студента к выполнению ЛР: наличие сценария, тестов, входной контроль знаний.
3. Демонстрация возможностей ОС UNIX на проекционном экране на лекции и на лабораторной работе.
4. Приобретение основных навыков работы в ОС UNIX на рабочих станциях в окнах текстовых терминалов (начало и окончание сеанса, информационные команды, просмотр и редактирование текстовых файлов, оперативная подсказка, механизм завершения имен *bash*, стек истории команд).
5. Навигация в иерархической файловой системе ОС UNIX (переход в домашний каталог, определение местонахождения, переходы вниз, вверх, вбок, в заданное место, посещение каталогов системы и других пользователей).
6. Манипуляции с файлами: создание, копирование, переименование и удаление файлов и каталогов.
7. Работа в текстовом редакторе *emacs* [2].
8. Сервисные функции: построение графиков, протоколирование сеансов и печать протокола сеанса лабораторной работы (минимальный состав протокола: **who, pwd, ls, cd, cp, mv, rm, cat, ps, mkdir, rmdir, uptime, rwho**).
9. Освоение оконных интерфейсов UNIX (X Window System) на X-терминалах или рабочих станциях.
10. Оформление отчёта установленной формы.
11. Защита отчёта со сдачей зачёта по материалам работы.

Сценарий работы:

Исторически доступ к ОС UNIX осуществлялся с помощью терминалов, подключенных к компьютеру по последовательным каналам (RS-232) непосредственно или через терминальный сервер, находящийся в локальной сети Ethernet. Такими терминалами сначала были электрические пишущие машинки с компьютерными интерфейсами (типа LA-120, до сих пор используемого в качестве принтера), а потом – текстовые строчные и экранные терминалы. В 1978 году (задолго до Mac и Windows!) ОС UNIX была снабжена оконным графическим интерфейсом через X-терминалы или непосредственно на рабочих станциях (имеются в терминальном классе). Терминальный доступ помимо аппаратного терминала возможен также в режиме эмуляции терминала с доступом по протоколам telnet, ssh и т. п., в том числе из MS Windows. Хорошим ssh-клиентом и эмулятором терминала удалённой ЭВМ в среде MS Windows является *putty*.

После включения текстового терминала, непосредственно соединённого с компьютером, на экране появляется заставка с именем узла сети, к которому подключен данный терминал. (Первый, кто угадает символ заставки, получит приз – 8-дюймовую дискету!). При использовании облачного терминала необходимо выбрать протокол и адрес соединения.

В настоящее время для доступа к ОС UNIX чаще используется рабочая станция, необходимо включить компьютер и дождаться загрузки ОС UNIX. Лабораторная система загружается автоматически по сети с сервера (для бездисковых рабочих станций) или с локального винчестера. В мультизагрузочной системе или плеере виртуальных машин необходимо выбрать соответствующую систему. Для доступа к серверам лаборатории из соседнего компьютерного класса необходимо выполнить сетевое соединение (ssh 172.16.0.255)

Начало сеанса ОС UNIX возможно в ответ на приглашение **Login:** или **Username:**. Следует ввести свое имя в системе (например 128123, совпадает с номером студбилета/зачётки), и пароль (в ответ на приглашение **Password:**, при этом вводимый пароль не отображается даже звёздочками). Пароли устанавливаются системным администратором для всех компьютеров сети, выдаются преподавателем на первом занятии и могут быть изменены самими студентами командой **kpasswd** (а не обычной **passwd**) в связи с тем, что для аутентификации пользователей применяется Kerberos). При смене необходимо указать старый пароль! Неуспех при входе в систему (сообщение **Login incorrect**) может означать ошибку как в пароле, так и в имени.

При успешном входе в систему на экране появляется т. н. рабочий стол пользователя в среде Gnome. Далее необходимо запустить эмулятор текстового терминала (Applications→Accessories→Terminal). Появление приглашения вида **пользователь@сервер:текущий-каталог\$** означает готовность системы к приёму и исполнению команд.

Переключиться в режим полноэкранного чисто текстового терминала можно с помощью комбинации клавиш Ctrl+Alt+F1, обратно — Ctrl+Alt+F9.

Сразу после входа в систему пользователь попадает в *собственную* (домашнюю) директорию, в которой и располагаются все его файлы и поддиректории. В лабораторной системе домашние директории студентов размещаются на серверах. При указании пути в файловой системе домашняя директория обозначается символом ~ (*тильда*). Можно воспользоваться командой **ls** с ключами **l**, **a** и **F** (**ls -laF**) для ознакомления с ее содержимым. В домашней директории также располагаются некоторые служебные файлы, описание которых выходит за рамки данной лабораторной работы. Заметим, что эти файлы видны по команде **ls** с соответствующей опцией (например, **-a**). Весьма полезен ключ **-F**, наглядно отображающий тип файла (выполнимый снабжается *, директория — /, символические ссылки — знаком @).

Выясните, какие узлы сети в настоящий момент доступны, а какие — нет (команда **ruptime**), а также, какие пользователи работают на всех доступных UNIX-машинах (**rwho**). Для определения имени пользователя, номера его группы, терминала, на котором ведётся работа, сетевого имени машины и используемой версии ОС UNIX соответственно применяются команды: **whoami** (или **who am i**), **tty**, **hostname** и **uname -a**, а для получения подробной информации о текущем пользователе — составная команда **finger `whoami`** (обратные апострофы означают подстановку вывода (результата) вложенной команды **whoami** в текст команды **finger** в качестве её аргумента). Результаты этих команд также используются для документирования печатного протокола сеанса. Например, **finger имя-пользователя** включает в протокол полное имя и другую информацию о пользователе.

Также можно попробовать другие полезные команды: **pwd** (для определения местонахождения в файловой системе; по этой команде отображается полный путь к текущей директории), **who** (для определения того, какие пользователи работают в данный момент на локальной машине), **man** (для получения оперативной документации по командам UNIX; наберите, например, **man ls** для просмотра всех ключей команды **ls**). Перейдите в корневую директорию UNIX (**cd /**), в директорию команд (**cd /bin**), просмотрите их оглавления, вернитесь в домашнюю директорию, перейдите вверх по дереву (**cd ..**) и убедитесь, что механизм защиты UNIX ограничивает доступ к директориям, владельцами которых являются другие пользователи.

После этого создайте в домашней директории новый файл: **cat > f1.txt**, введите несколько строк на терминале, завершив их нажатием **Ctrl-d** в начале строки (ввод буферизуется!), и убедитесь, что в домашней директории создан файл **f1.txt**, содержащий введённые строки. Для просмотра этого файла введите команду **cat f1.txt**. Для редактирования файла можно воспользоваться текстовым редактором Emacs, для этого введите команду **emacs f1.txt**. Подробное описание возможностей Emacs содержится в соответствующем справочном материале, прилагаемом к данному руководству. Переключение на кириллицу в среде Gnome осуществляется по **Alt-Shift**. Выход из Emacs с сохранением внесённых изменений — **Ctrl-X S Ctrl-X Ctrl-C** (или **Ctrl-X Ctrl-C y**).

Скопируйте файл **f1.txt** в файл **f2.txt** с помощью команд **cp** или **cat**. Сконкатенируйте файлы **f1.txt** и **f2.txt** в файл **f3.txt**. Просмотрите содержимое полученного файла на терминале и убедитесь в правильности выполнения конкатенации. Удалите файлы **f2.txt** и **f3.txt** командой **rm**.

Комбинации **ls** и **cat** позволяют отражать состояние файлов и директорий до и после различных операций в текстовом протоколе сеанса.

Для освоения средств работы с директориями создайте в своей домашней директории подкаталоги **lab** и **course**. Скопируйте файл **f1.txt** в эти поддиректории и удалите исходный файл. Перейдите в одну из поддиректорий, убедитесь в наличии скопированного файла и удалите его. Вернитесь в домашнюю директорию и попытайтесь удалить директории. Объясните причину неудачи и решите проблему.

Для демонстрации средств графики вызовите утилиту **gnuplot** для построения двух- и трехмерных графиков.

```
пользователь@сервер:директория> gnuplot
```

```
...
gnuplot> plot sin(x)*cos(x)
...
```

Попробуйте выполнить параметрические построения окружности и сферы:

```
gnuplot> set parametric
gnuplot> set trange [0 to 2*pi]
gnuplot> set xrange [-1 to 1]
gnuplot> set yrange [-1 to 1]
gnuplot> plot sin(t),cos(t)
gnuplot> set zrange [-1 to 1]
gnuplot> set urange [0 to 2*pi]
```

```
gnuplot> splot sin(x*y*y)
```

```
...
gnuplot> set vrange [0 to 2*pi]
gnuplot> set nohidden3d
gnuplot> splot sin(u)*sin(v),sin(u)*cos(v),cos(u)
gnuplot> set noparametric
...
gnuplot> exit
```

Более подробная информация по **gnuplot** содержится в собственных средствах помощи утилиты (встроенная команда **help**). Свободно распространяемая утилита **gnuplot** реализована практически на всех современных системах и широко применяется научными работниками и инженерами. Эта программа весьма полезна для выполнения и оформления учебных работ по математическому анализу. Интересно, что **gnuplot** может работать даже на пишущей машинке! Для этого в текстовом окне используйте **gnuplot** с опцией **dumb** (**set term dumb**) и вы увидите пример т. н. ASCII-графики — вывод графики в текстовом режиме.

Дополнительно рекомендуется продемонстрировать такие замечательные возможности UNIX как конвейеры (команды – цепочки элементарных фильтров, конкатенируемых с помощью вертикальной черты), метасимволы (*, ?, [], &), цепочки команд (через точку с запятой).

Механизм завершения имен **bash** позволяет, находясь в командной строке, нажатием клавиши **Tab** достраивать, в зависимости от контекста, имена файлов, команд и их опций. Клавишами \uparrow и \downarrow функциональной клавиатуры можно передвигаться по стеку ранее введенных командных строк, а \leftarrow и \rightarrow — перемещаться по выбранным из истории сеанса строкам для их корректировки и последующего выполнения. Полезно вставлять в историю команд заготовки в форме комментариев и искать их в дальнейшем с помощью **Ctrl-r**.

После освоения основных команд UNIX следует продемонстрировать это преподавателю, и выполнить демонстрационный сеанс в режиме протоколирования. Протоколирование представляет собой запуск еще одного (вложенного) командного интерпретатора UNIX, ввод и вывод которого дублируется во временный файл и через серию конвейеров направляется в очередь печати на сетевой принтер. В состав сеанса должны входить основные команды (**date, who, pwd, ls, cd, cp, mv, rm, rwho, ruptime, cat**). Сеанс протоколирования начинается командой **proto**. НЕЛЬЗЯ использовать в сеансе протоколирования программы, дающие *нетекстовый* вывод (**emacs, gnuplot, man, tu, jdt, nam, cat** нетекстового файла и др.), иначе протокол будет испорчен. Протоколирование завершается командой выхода из вновь запущенного интерпретатора команд UNIX **exit**, после чего возможен отказ от выдачи на печать неудачного протокола. Объем протокола, как правило, не должен превышать 164 строк (1 листа формата A3, на котором библингом размещаются 2 страницы формата A4) чтобы уложиться в нормы расхода бумаги. При электронном документировании работ протокол заверяется электронной подписью студента и преподавателя, заносится в архив и реестр электронных документов. Тестирование и протоколирование производится только на серверах неинтеловской архитектуры (DEC Alpha).

x x x

Важной особенностью ОС UNIX является оконная система X Window System, созданная еще в 1978 г. Эта система является не просто оконной оболочкой ОС, но полноценной распределенной сетевой оконной системой, обеспечивающей прозрачную работу графических приложений по сети, поддержку специальных оконных приложений и подключение к базовому компьютеру по сети дополнительных графических терминалов (X-терминалов) включая распределённый рендеринг OpenGL-сцен. Чрезвычайная гибкость системы обусловлена тем, что в отличие от большинства настольных систем (MS Windows) логика работы системы отделена от функций отображения, и может выполняться на разных ЭВМ (при этом использована распределенная технология клиент-сервер). В ходе работы необходимо изучить и освоить основные возможности X Window System и сравнить ее с другими известными оконными системами ПЭВМ: Microsoft Windows, Apple MacOS (X) (по литературе, за X-терминалом, на рабочих станциях ОС UNIX (Intel) или на домашних компьютерах).

Замечание. Сценарий работы в ОС UNIX можно подготовить путем написания командного файла (скрипта) на shell, который предусматривает все необходимые действия: создает файлы, копирует, перемещает и т. д. Скрипт — это выполнимый (интерпретатором командного языка ОС UNIX) текстовый файл, в котором строка за строкой записаны необходимые команды, например:

```
#!/usr/bin/env bash
echo "Hello, world!" > f1.txt
cp f1.txt f2.txt
mkdir d1 d2 d1/d11
# ...
```

Запустить такой файл можно следующей командой:

```
./lab2.sh
```

Установить для текстового файла признак выполнимости можно командой **chmod**.

Для ввода данных во входном потоке данного командного файла необходимо использовать конструкцию

```
cat << EOF > file
bla-bla-bla
...
end-of-bla-bla-bla
EOF
```

Применение скрипта не освобождает студента от необходимости приобретения мануальных навыков работы в командной строке UNIX.

Более подробное изучение интерпретируемого командного языка ОС UNIX будет продолжено в рамках лабораторных работ № 3, 20, 21.

Для успешного выполнения работы студенту рекомендуется составить сценарий работы, посетить поточные и групповые консультации, потренироваться на свободно распространяемой версии UNIX. Это можно сделать и в среде MS Windows на эмуляторах UNIX (Cygwin, Uwin, Interix и т. п.). Для допуска к работе №2 студент *должен* иметь первую часть отчета по лабораторной работе на бланке (пункты 1–7 правил оформления лабораторных работ), по которому может быть проведен входной контроль знаний. Для защиты работы №2 необходимо подготовить отчет, включая подписанный преподавателем листинг, и сдать промежуточный зачет по балльно-рейтинговой системе. Материалы ЛР №2 излагаются на лекции по практикуму и на лабораторной работе.

ОПЕРАЦИОННАЯ СИСТЕМА. ОС UNIX.

1. Понятие о программном обеспечении ЭВМ. Состав, структура, многоуровневый иерархический характер.
2. Понятие об операционной системе.
3. Краткая историческая справка об ОС UNIX.
4. Особенности ОС UNIX (режим работы, инструментальные и сервисные средства, язык Си, TCP/IP, Internet и пр.).
5. Основы интерактивной работы в ОС UNIX. Начало и окончание сеанса. Доступ к оперативной документации.
6. Команды ОС UNIX. Формат команд. Интерпретатор команд.
7. Протоколирование сеансов ОС UNIX.
8. Файловая система ОС UNIX (структура, именование, навигация).
9. Метасимволы для генерации имен файлов.
10. Управление файлами (создание, копирование, сцепление, переименование, удаление).
11. Управление каталогами (создание, просмотр, удаление).
12. Переадресация ввода и вывода в ОС UNIX.
13. Конвейеры команд.
14. Фоновые процессы: запуск, остановка, слежение. Приостановка и возобновление emacs.
15. Понятие о программировании на ИКЯ (Shell, Cshell, bash...): переменные, параметры, выражения, присваивание, проверки, ветвления, циклы

РЕДАКТИРОВАНИЕ ТЕКСТОВ

1. Понятие о редакторе текстов. Основные типы редакторов текстов (строчный, экранный, графический, пакетные и текстовые процессоры).
2. Строчное редактирование. Концепция. Пример: основные возможности стандартного строчного редактора ОС UNIX **ed**.
3. Экранное редактирование. Концепция. Примеры экранных редакторов ОС UNIX (**vi**, **vim**, **emacs**, **ted**).
4. Краткая историческая справка по редактору emacs. Особенности реализации.
5. Базовые возможности emacs (переход между кириллицей и латинским алфавитом, выход с сохранением всех измененных буферов и без сохранения, завершение имен, откат и перерисовка).
6. Меню редактора: вызов, использование и отмена меню. Виды меню. Структура главного меню.
7. Перемещение курсора в emacs.
8. Удаление текста в emacs.
9. Работа с блоками текста в emacs.
10. Поиск и замена в emacs.
11. Работа с буферами и окнами в emacs.
12. Операции с файлами в emacs.
13. Вызов интерпретатора команд UNIX из emacs.

Литература

1. Готье, Руководство по ОС UNIX, –М.: Финансы и статистика, 1985.
2. Экранный редактор текстов emacs. ШПАРГАЛКА пользователя. -М.: МАИ, 1997. - 2 с.
3. Бек Л. Введение в системное программирование –М.: Мир, 1988. с.411–429.
4. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX.– БХВ, 2010.– 656 с. ISBN 978-5-94157-538-1
5. Беляков М.И. и др. Мобильная операционная система. -М.: Радио и связь, 1991. с. 155-157, 161-164.
6. Цикритзис Д., Бернштейн Ф. Операционные системы. - М.: Мир, 1977.
7. Танненбаум Э. Многоуровневая организация ЭВМ. -М: Мир, 1979
8. Дейтел Г. Операционные системы. - М.: Мир, 1987. Том 2, тт. 16.1,2,3,10,11,12, гл. 18,19
9. Кристиан К. Введение в операционную систему UNIX. –М.:Финансы и статистика, 1985.
10. Баурн С. Операционная система UNIX. –М.:Мир, 1986.
11. Официальный сайт проекта GNU: <http://www.gnu.org>
12. МакМаллен Дж., UNIX (Complete Idiot's Guide) // Пер. с англ. –М.: ЮНИТИ, 1996.
13. Браун П., Введение в операционную систему UNIX, –М.: Мир, 1987.
14. Банахан М., Раттер Э. Введение в операционную систему UNIX. –М.: Радио и связь, 1986.
15. Керниган Б.В., Пайк Р. UNIX – универсальная среда программирования. –М.: Финансы и статистика, 1992.
16. Инструментальные средства разработки программ в современных операционных системах // Учеб. пособие под ред. С.М.Юдина. - М.: Изд-во МАИ, 1990, с. 38-53, 78-85 (интерпретатор команд cshell).
17. Топхем Д., Хай Ван Чыонг. Юникс и Ксеникс: Пер. с англ. – М.: Мир, 1988.
18. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования. – М.: Статистика, 1988. (C Shell, make, lex, yacc).
19. Зайцев В.Е., Сошников Д.В. и др. CD-хрестоматия по курсу Информатика. – М.: 2002 – в 2-х тт. на CD-ROM.
20. Жоголев Е.А. Лекции по технологии программирования. – М.: ИО ф-та ВМиК МГУ, 2001.(ISBN 5-89407-008-2) Главы 2,10. см. также <http://www.cmc.msu.ru>