
**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ»

Кафедра автоматизированных систем управления (АСУ)

УТВЕРЖДАЮ

Зав. кафедрой АСУ, профессор



А.М. Корилов

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ

Лабораторная работа №1: «Подготовка работы на УПК АСУ»

Учебно-методическое пособие

для студентов уровня основной образовательной программы магистратура
направления подготовки 010400.68 «Прикладная математика и информатика»
профиля Математическое и программное обеспечение вычислительных комплексов и
компьютерных сетей

Разработчик

доцент кафедры АСУ

В.Г. Резник

Резник В.Г.

Архитектура вычислительных комплексов. Лабораторной работы №1: Подготовка работы на УПК АСУ. Учебно-методическое пособие. – Томск, ТУСУР, 2014. – 29 с.

Учебно-методическое пособие предназначено для выполнения лабораторной работы №1 по дисциплине «Архитектура вычислительных комплексов» студентов уровня основной образовательной программы магистратура направления подготовки 010400.68 «Прикладная математика и информатика» профиля «Математическое и программное обеспечение вычислительных комплексов и компьютерных сетей».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. СТРУКТУРА УПК АСУ	5
1.1. Структура дистрибутива ОС Xubuntu	5
1.2. Структура директорий УПК АСУ	7
1.3. Подготовка личного flash-накопителя	8
2. ПРАКТИЧЕСКАЯ РАБОТА СО СРЕДОЙ УПК АСУ И КЛАСТЕРА	10
2.1. Настройка сетевого обеспечения ОС	10
2.2. Консольный доступ к кластеру	12
2.3. Архитектура X Window System	15
3. КОНТРОЛЬ ВЫПОЛНЕНИЯ РАБОТ	22
3.1. Удаленная разработка приложений на базе IDE Eclipse	22
3.2. Контроль выполнения лабораторной работы №1	26
ЛИТЕРАТУРА	28

ВВЕДЕНИЕ

Рассматриваемое учебно-методическое пособие содержит описание ряда технологических навыков, которые студент должен освоить для выполнения лабораторных работ по дисциплине «АВК - Архитектура вычислительных комплексов».

Тема лабораторной работы: «Подготовка работы на УПК АСУ».

Изложенный материал является обязательной частью обучения, поскольку содержит информацию для выполнения последующих лабораторных работ по указанной дисциплине.

Последовательность и изложение материала данного пособия предполагает, что магистрант:

- прошел обучение по курсу «Современные операционные системы»;
- имеет на собственном flash-носителе загрузчик ОС УПК АСУ;
- умеет запускать ОС УПК АСУ, как в текстовом, так и в графическом режимах;
- имеет личный защищенный доступ к вычислительному кластеру кафедры АСУ.

Изложенный материал разбит на три раздела, которые определяют последовательность процесса обучения.

В первом разделе дается описание структуры программного обеспечения УПК АСУ применительно к тематике всего курса.

Во втором разделе, последовательно излагаются действия:

- по настройке сетевого обеспечения ОС рабочей станции студента;
- правила установки защищенного терминального соединения с кластером ЭВМ кафедры АСУ;
- особенности архитектуры X Window System, а также параметры доступа, обеспечивающие возможность подключения программ, запущенных на кластере, к собственному X-серверу.

Третий раздел описывает требования по контролю выполняемых работ.

Замечание 1.

Технология проведения всех работ, предполагает, что каждый студент должен иметь flash-память не менее 2 Гб для сохранения личного материала и запуска УПК АСУ.

1. СТРУКТУРА УПК АСУ

Данная часть лабораторной работы содержит описание среды ОС, в которой проводятся лабораторные работы. Такая среда содержит:

- компьютеры учебных классов кафедры АСУ, полностью обеспечивающие индивидуальное обучение магистранта;
- общее программное обеспечение каждого компьютера с установленной ОС MS Windows XP, подключенной к общей локальной сети кафедры АСУ;
- специальное программное обеспечение УПК АСУ, содержащее обучающие программы и учебно-методический материал по изучаемой дисциплине.

Для выполнения данной и последующих лабораторных работ, магистрант должен изучить:

- местоположение и файловую структуру УПК АСУ в пределах файловой системы ОС MS Windows XP;
- местоположение и состав учебно-методического материала данной и последующих лабораторных работ;
- местоположение и состав дистрибутива ОС Xubuntu;
- местоположение и состав учебных вариантов ОС УПК АСУ;
- подготовить личный накопитель flashUSB для выполнения лабораторных работ и хранения личных данных, включающих загрузочный вариант ОС и общий индивидуальный отчет по выполненным работам.

1.1. Структура дистрибутива ОС Xubuntu

Являясь свободным программным обеспечением, Linux всегда вызывал интерес широкого круга пользователей. Препятствием этому интересу служила необходимость дополнительной настройки конфигурации ОС, что требовало не только затрат времени, но и наличия специальных знаний. Прежде всего, это казалось:

- поддержки национальных языков;
- настройки графических режимов адаптеров ЭВМ, сравнимых по качеству с широко известной ОС MS Windows.

В 2004 году, сообщество разработчиков Linux, возглавляемое компанией Canonical, выпустила дистрибутив Ubuntu, основанный на дистрибутиве Debian.

Удачное начало работы Canonical породило целую серию направлений, которые специализируются на поддержке различных графических сред, реализованных в рамках различных проектов Linux.

Используемый для обучения дистрибутив Xubuntu 14.04 LTS обладает следующими качествами:

- 32-битный дистрибутив обеспечивает работу на компьютерах кафедры АСУ и может быть установлен на 64-битные ЭВМ;
- графический интерфейс Xfce не требователен к ресурсам графических адаптеров ЭВМ и обеспечивает устойчивую работу в конфигурациях LiveCD (LiveUSB);
- версия LTS (Long Term Support — долгосрочная поддержка), обеспечивает актуальность и современность ОС в течение 5 лет, начиная с апреля 2014 года.

Дистрибутивы Xubuntu обычно представлены файлами в формате iso9660 и предназначены для записи их на диски CD-ROM (DVD-ROM). Скачать такие файлы, в зависимости от типа процессора, можно со страницы Canonical Ltd.: <http://cdimage.ubuntu.com/xubuntu/releases/trusty/release/>.

Например, используемый в УПК АСУ файл **xubuntu-14.04-desktop-i386.iso** имеет размер 942.7 Мб. Распаковать его в MS Windows можно, например, утилитой **WinRAR**.

Среди множества директорий, входящих в архив дистрибутива, особый интерес представляет директория **casper**, которая содержит файлы:

vmlinuz	Сжатое ядро ОС (5800.848 Кб).
initrd.lz	Сжатая временная файловая система , которая предназначена для загрузки основных модулей ОС и монтирования корневой файловой системы (20398 Кб).
filesystem.squashfs	Сжатая корневая файловая система , предназначенная как для работы ОС в режиме LiveCD, так и для инсталляции дистрибутива (887.540 Мб).

Фактически, эти три файла и нужны для работы современного дистрибутива ОС Linux.

Замечание 2.

Дистрибутивы Ubuntu, для загрузки ядра **vmlinuz** и файловой системы **initrd.lz**, используют специальный загрузчик GNU GRUB, входящий в дистрибутив самой ОС.

Предполагается, что магистрант уже изучил технологию загрузки ОС Xubuntu с помощью загрузчика GRUB.

2.2. Структура директорий УПК АСУ

На ЭВМ в компьютерных классах кафедры АСУ установлены ОС MS Windows XP. Магистр, проходящий обучение по изучаемой дисциплине «Архитектура вычислительных комплексов» должен:

- иметь личные данные для авторизации на компьютерах кафедры АСУ;
- уметь включить компьютер и пройти процедуру авторизации;
- знать место хранения личных данных на серверах кафедры, а также иметь личный flash-накопитель и уметь пользоваться им.

Блочные устройства MS Windows имеют буквенные обозначения: **a:**, **b:**, ..., **z:**.
Корневая файловая система ОС расположена на устройстве **c:**.

Все программное обеспечение учебного комплекса УПК АСУ находится в директории [c:\asu14.04upk](#). Содержимое учебного и программного материала расположено в директориях, перечисленных в табл. 1.1. Следует изучить и запомнить назначение этих директорий, поскольку они используются в процессе всего курса обучения.

Таблица 1.1. Назначение директорий УПК АСУ по данному курсу обучения

Директория	Содержимое директории
<i>casper</i>	<i>vmlinuz</i> — ядро ОС, используемое во всех вариантах запуска ПО УПК АСУ.
<i>upk</i>	<i>initrd2</i> — временная файловая система для запуска ОС в варианте УПК АСУ; <i>upkfs.sfs</i> — корневая ФС дистрибутива УПК АСУ.
<i>avk-010400.68</i>	Директория учебно-методического материала по курсу «Современные компьютерные технологии». Содержит директории <i>lab1</i> , <i>lab2</i> , ..., <i>lab5</i> с необходимым учебным материалом и ПО.
<i>opt</i>	Директория инструментальных пакетов программ, которые необходимы для выполнения лабораторной работы: <i>eclipseC.sfs</i> — инструментальная среда разработки ПО на языках C и C++ для лабораторных работ, выполняемых на локальном компьютере.

1.3. Подготовка личного flash-накопителя

Магистрант должен иметь личный загрузочный flashUSB с установленным ПО GRUB.

Предполагается, что *личный flashUSB* имеет структуру меню запуска, аналогичную структуре, приведенной на листинге 1.1.

Листинг 1.1. Содержимое файла grub.cfg

```
###
# Reznik, 02.08.2014
# BEGIN /etc/grub.d/40_custom
###
#
GRUB_HIDDEN_TIMEOUT_QUIET=false
GRUB_CMDLINE_LINUX_DEFAULT=""
GRUB_CMDLINE_LINUX=""
GRUB_TERMINAL=console
GRUB_DISABLE_LINUX_UUID=true
#
set pager=1
#
set lang="ru_RU"
loadfont "/boot/grub/fonts/unicode.pf2"
#
insmod vga
insmod gettext
terminal_output gfxterm
#
#
set commonpar=" asuuser=vgr asudomain=asu asugroup=g432-1 asulive "
#
menuentry "asu14.04upk на диске HD1 файл upkfs.sfs"{
    insmod gzio
    insmod part_msdos
    insmod ntfs
    insmod ext2
    set root='(hd1,1)'

    set par1=" boot=casper ro locale=ru_RU "
    set par2=" hdboot=asu14.04upk/upk hddev=/dev/sda1 hdtype=ntfs "

    linux /asu14.04upk/casper/vmlinuz ${par1} ${commonpar} ${par2}
    initrd /asu14.04upk/upk/initrd2
}
#
menuentry "Linux-busybox-Xubuntu14 на личном flashUSB"{
    insmod gzio
    insmod part_msdos
    insmod ntfs
    insmod ext2
    set root='(hd0,1)'

    linux /casper/vmlinuz ro locale=ru_RU hostname=vgr shell=/bin/sh
    initrd /casper/initfs.lz
}
```



```
#
menuentry "Linux-busybox-Xubuntu14 on sda1"{
    insmod gzio
    insmod part_msdos
    insmod ntfs
    insmod ext2
    set root='(hd1,1)'

linux /asul4.04upk/casper/vmlinuz ro locale=ru_RU hostname=vgr shell=/bin/sh
initrd      /asul4.04upk/upk/initfs.lz
}
#
menuentry "MS Windows (loader) on /dev/sda1" --class windows --class os {
    insmod part_msdos
    insmod ntfs
    set root='(hd1,msdos1)'
    chainloader +1
}
#
### END /etc/grub.d/40_custom ###
```

Замечание 3.

Индивидуальные настройки меню GRUB, приведенные на *листинге 1.1*, должны быть **согласованы с преподавателем**.

В корне файловой системы личного flashUSB должна находиться директория **archives** с файлами, которые можно найти в директории **.../avk-010400.68/lab1** на рабочих станциях учебных классов УПК АСУ.

Для выполнения лабораторных работ, обязательным является наличие файлов, перечисленных в таблице 1.2.

Таблица 1.2. Файлы директории *archives* личного flashUSB обучающегося

Файл	Содержание
startC	Сценарий запуска среды разработки Eclipse для разработки программ на языках C и C++.
tar_avk	Сценарий сохранения архива директории /home/upk на личном flashUSB в директории archives . Следует обязательно выполнять после завершения лабораторной работы!!!
untar_avk	Сценарий восстановления директории /home/upk из архива на личном flashUSB, который находится в директории archives . Следует обязательно выполнять после старта ОС и перед началом выполнения лабораторной работы!!!

2. ПРАКТИЧЕСКАЯ РАБОТА СО СРЕДОЙ УПК АСУ И КЛАСТЕРА

Сетевое ПО кафедры АСУ настроено таким образом, что компьютеры получают IP-адрес автоматически по протоколу ДНСР. При старте ОС УПК АСУ использует протокол ДНСР для получения адреса, маски и основного маршрутизатора компьютера.

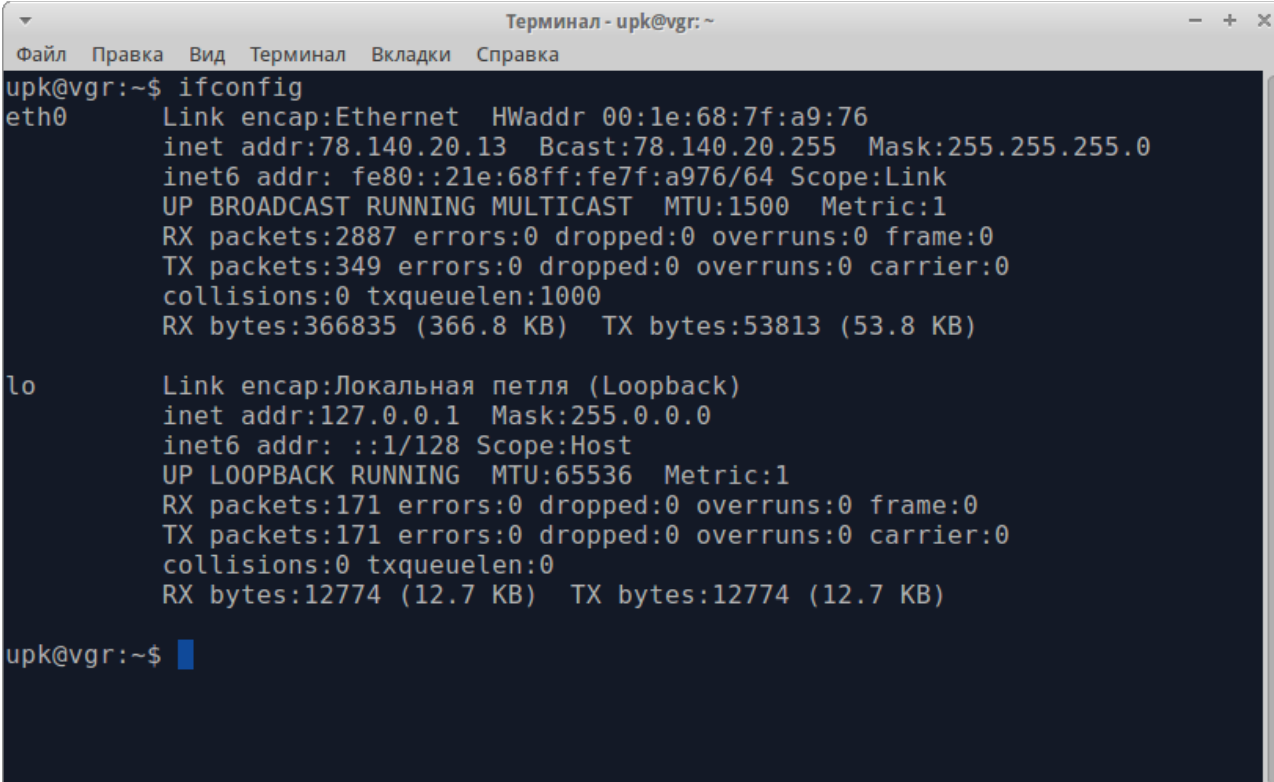
Замечание 4.

Если при запуске ОС УПК АСУ сеть не установилась, следует обратиться к преподавателю, обеспечивающему проведение лабораторных работ.

2.1. Настройка сетевого обеспечения ОС

Проверка работоспособности настроек сети проводится разными способами:

- запустив из меню рабочего стола «Эмулятор терминала», можно выполнить команду *ifconfig*; тогда появится информация, например, как показано на рисунке 2.1; сообщение, выделенное как *eth0*, покажет IP адрес компьютера, его маску и другие параметры;



```
Терминал - upk@vgr: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
upk@vgr:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1e:68:7f:a9:76
          inet addr:78.140.20.13  Bcast:78.140.20.255  Mask:255.255.255.0
          inet6 addr: fe80::21e:68ff:fe7f:a976/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2887 errors:0 dropped:0 overruns:0 frame:0
          TX packets:349 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:366835 (366.8 KB)  TX bytes:53813 (53.8 KB)

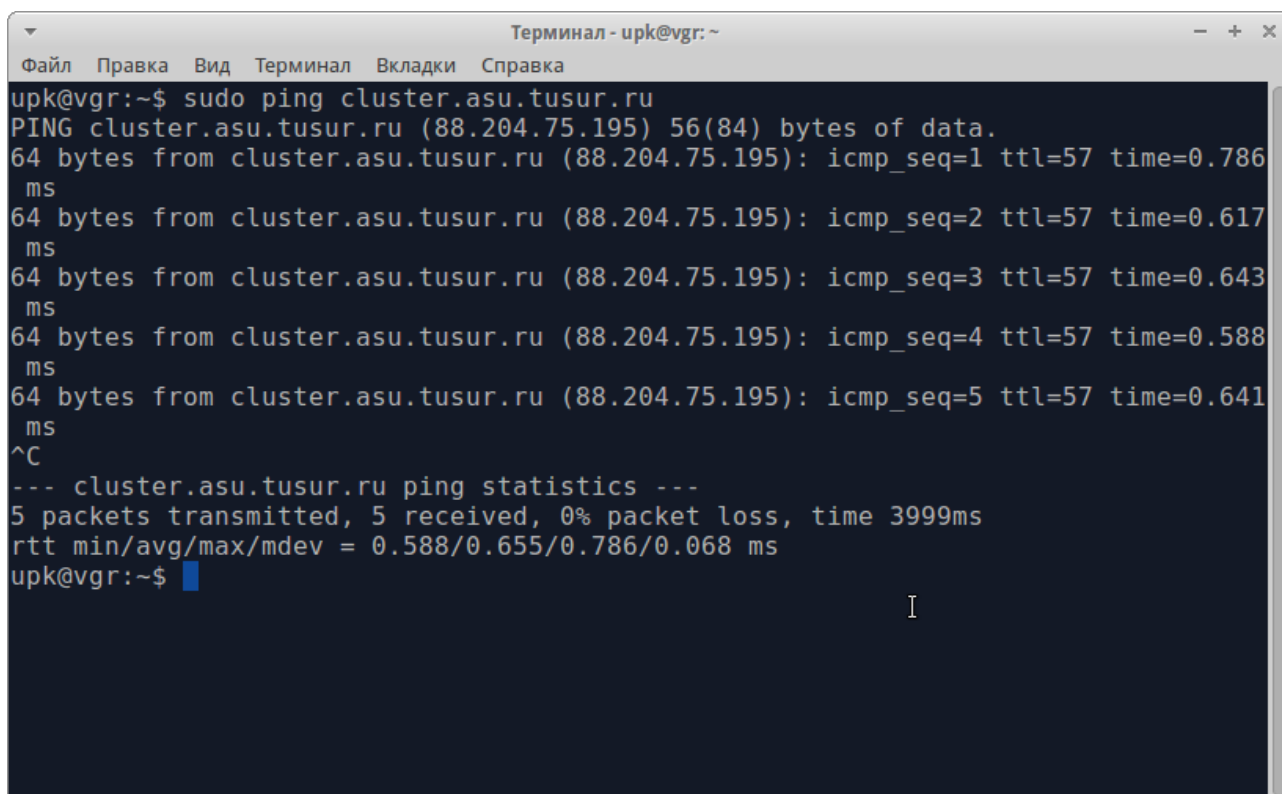
lo        Link encap:Локальная петля (Loopback)
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:171 errors:0 dropped:0 overruns:0 frame:0
          TX packets:171 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12774 (12.7 KB)  TX bytes:12774 (12.7 KB)

upk@vgr:~$
```

Рис. 2.1. Сообщения команды ifconfig

- убедившись подключении рабочей станции к сети, следует проверить доступность кластера ЭВМ кафедры АСУ; для этого необходимо, как показано на рисунке 2.2, выполнить командой **ping** по адресу кластера:

sudo ping cluster.asu.tusur.ru



```
Терминал - upk@vgr: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
upk@vgr:~$ sudo ping cluster.asu.tusur.ru
PING cluster.asu.tusur.ru (88.204.75.195) 56(84) bytes of data.
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=1 ttl=57 time=0.786
ms
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=2 ttl=57 time=0.617
ms
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=3 ttl=57 time=0.643
ms
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=4 ttl=57 time=0.588
ms
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=5 ttl=57 time=0.641
ms
^C
--- cluster.asu.tusur.ru ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.588/0.655/0.786/0.068 ms
upk@vgr:~$
```

Рис. 2.2. Проверка наличия кластера командой ping

Убедившись, что кластер обнаружен, необходимо остановить работу **ping**, используя комбинацию клавиш **<Ctrl>+C**.

Замечание 5.

Если кластер ЭВМ не обнаруживается, следует обратиться к преподавателю, обеспечивающему проведение лабораторных работ.

2.2. Консольный доступ к кластеру

Учебная цель данного этапа, научиться устанавливать защищенное соединение с кластером и настраивать личные переменные среды окружения удаленной ОС.

Для обеспечения такого доступа к удаленному компьютеру, существует множество программ, как для ОС MS Windows, так и для ОС Linux. Наиболее известной из них является программа PuTTY, которая имеется для обеих ОС.

Мы рассмотрим наиболее простой способ, который осуществляется с помощью «Эмулятора терминала» ОС Linux посредством вызова программы **ssh** (*security shell*). Чтобы выполнить такое подключение, необходимо набирать команду:

```
ssh cluster.asu.tusur.ru -p 6022 -l <имя пользователя>
```

где **cluster.asu.tusur.ru** - адрес главного компьютера кластера кафедры АСУ;
-p 6022 - номер порта подключения к кластеру;
-l <имя пользователя> - имя пользователя, под которым он имеет регистрацию на сервере кластера.

Пример такого действия показан на следующем рисунке 2.3:

```

ms
64 bytes from cluster.asu.tusur.ru (88.204.75.195): icmp_seq=5 ttl=57 time=0.641
ms
^C
--- cluster.asu.tusur.ru ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.588/0.655/0.786/0.068 ms
upk@vgr:~$ ssh cluster.asu.tusur.ru -p 6022 -l reznik
Password:
Last login: Tue Aug 26 17:20:00 2014 from n20-cl3.client.tomica.ru
Have a lot of fun...
##### My set #####
##### set Report #####
avkReport=libreoffice /home/asu/reznik/reports/avk-432-m/reznik.odt
##### set Eclipse #####
myIDE=/home/asu/reznik/eclipsePTP/eclipse -data /home/asu/reznik/workspace
##### set DISPLAY #####
DISPLAY=78.140.20.13:0
#####
reznik@tusur-master:~>

```

Рис. 2.3. Пример терминального соединения с кластером

Замечание 6.

Удаленный сервер может запросить согласие на генерацию ключей. Нужно ответить: **yes**.

Затем сервер запросит пароль, который студент должен правильно ввести и нажать клавишу «**Enter**».

Замечание 7.

Имя и пароль для доступа на кластер кафедры АСУ студент получает у преподавателя, ведущего лабораторные занятия.

Если студент неправильно введет пароль, то удаленный сервер предложит повторный ввод. После нескольких неудачных попыток, сервер разрывает соединение.

Если пароль введен правильно, сервер выдает подсказку, характерную для консоли (терминала) ОС.

Далее, командой **mc** следует запустить Midnight Commander и изучить свою (домашнюю) директорию пользователя, как показано на рисунке 2.4.

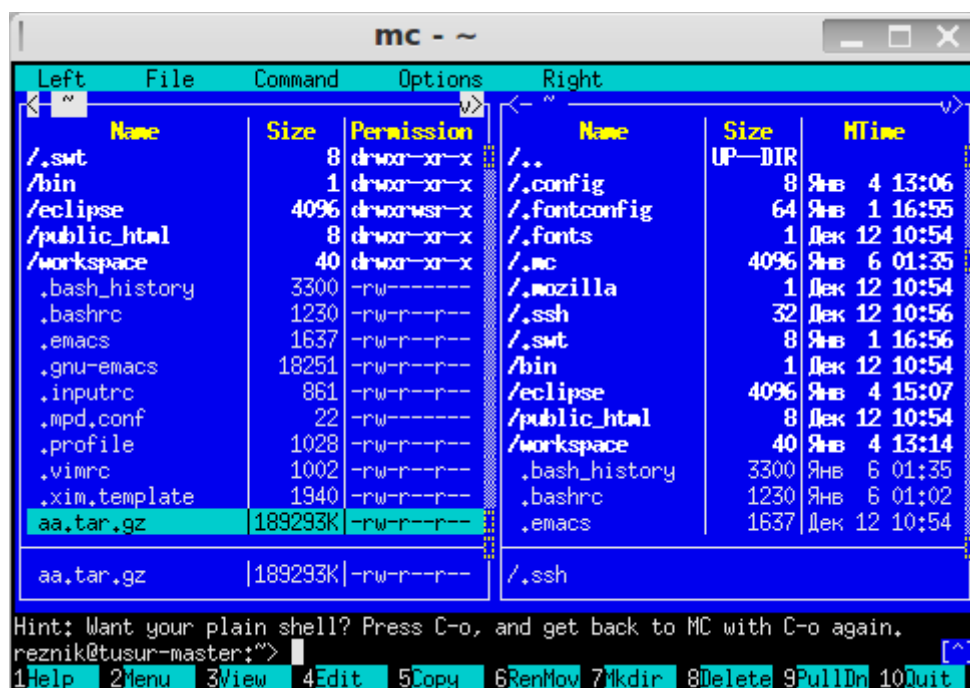


Рис. 2.4. Домашняя директория пользователя на кластере кафедры АСУ

Наиболее важными для изучения являются файлы сценариев **.profile** и **.bashrc**. Эти сценарии **запускаются**, когда пользователь «**успешно сделал login**» в системе, и **создают** переменные окружения (переменные среды) ОС.

Наиболее важными являются переменные, представленные в табл. 2.1.

Таблица 2.1. Наиболее важные переменные среды ОС

DISPLAY	Указывает адрес и экран <i>локальной</i> или <i>удаленной ЭВМ</i> , на которую буде осуществляться графический вывод приложений. <i>Подробнее рассмотрена в следующем подразделе.</i>
HOME	Директория пользователя.
PATH	Список путей, в которых ищутся запускаемые программы.
USER	Имя пользователя, под которым осуществлен вход в систему.

Чтобы проверить значения этих переменных, можно воспользоваться командами:

```
echo $USER<Enter>
echo $<Enter>
echo $PATH<Enter>
```

В большинстве случаев, достаточно редактирования файла **.bashrc**. Закончив его редактирование, необходимо выполнить команду:

```
. ~/.bashrc
```

Затем, следует повторно проверить значения этих переменных и завершить удаленный доступ к кластеру.

Замечание 8.

Закончив изучение файловой системы удаленного компьютера, студент должен корректно выйти из удаленного соединения (**просто закрывать окно терминала не следует!**).

Для этого, обычно достаточно последовательности команд **exit**.

Например, в рассмотренном случае достаточно двух команд **exit**:

- **первая** — закроет приложение Midnigth Commander;
- **вторая** — разорвет соединение с удаленным сервером.

2.3. Архитектура X Window System

Учебная цель этого подраздела:

- студент изучить архитектуру X Window System;
- научиться запускать X-сервер с нужными параметрами доступа;
- проверить возможность подключения к X-серверу программ, запущенных на сервере кластера.

Замечание 9.

В отличие от MS Windows, имеющей встроенное в ядро ОС графическое ПО, без которого она не может работать, Linux для этой цели имеет набор приложений, взаимодействующих на основе модели «клиент/сервер».

*Базовая часть графического ПО Linux называется **X Window System** (или просто - **X Window**).*

***Начиная с 1988 года**, этот стандарт поддерживался консорциумом X, созданным с целью унификации графического интерфейса для ОС UNIX.*

***В 1997 году** консорциум X был преобразован в X Open Group (<http://www.x.org>).*

X Window — достаточно сложная система и подробно описана большим количеством первоисточников. Мы рассмотрим только основные элементы архитектуры этой системы, опираясь на материал Интернет-статьи «Графический интерфейс Linux» [1].

Согласно этой статье, общая архитектура X Window может быть представлена рисунком 2.5.

«Сердцем» этой графической системы является программа X-сервер, который через драйверы устройств взаимодействует с видеоплатой, клавиатурой, мышью и монитором компьютера. Именно X-сервер устанавливает и переключает графические режимы видеоплаты, рисует элементы изображений, определяет координаты мыши и формирует программные прерывания при нажатии кнопок мыши и клавиатуры. Все остальные программы, включая менеджер окон, взаимодействуют с X-сервером по особому протоколу, который называется X-протокол, или протокол сетевой связи (X Network Protocol).

Для написания программ, поддерживающих **X-протокол**, имеется базовая библиотека **X-lib**. На основе этой библиотеки пишутся дополнительные графические библиотеки более высокого уровня, например, **GTK+**, **Qt**, **Motif** и другие. Обычно, менеджеры окон, рабочие столы и сложные графические приложения пишутся с использованием этих библиотек.

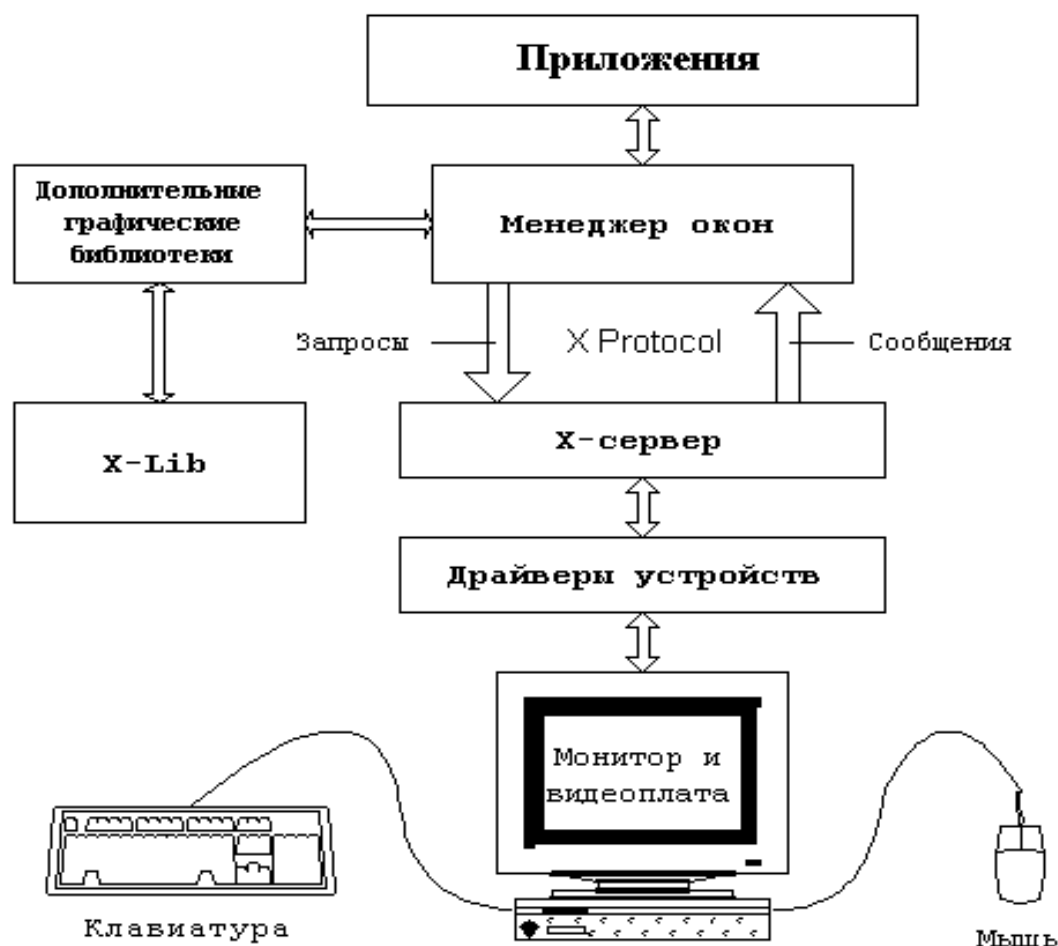


Рис. 2.5. Общая архитектура X Window

В общем случае, X-сервер обрабатывает **4 типа сообщений**:

- **Запрос** – клиент требует нарисовать что-либо в окне или запрашивает у сервера информацию;
- **Ответ** – сервер отвечает на запрос;
- **Событие** – сервер сообщает клиенту о событии (например, о нажатии клавиши пользователем);
- **Ошибка** – сервер сообщает об ошибке.

Замечание 10.

Важной для нас особенностью является возможность X-сервера работать на стеке протоколов TCP/IP с программами, запущенными на удаленных компьютерах. Обычно используется асинхронная связь (**протокол UDP**), но возможна и синхронная связь по **протоколу TCP**, которая работает в 30 раз медленнее.

Кроме того, на компьютере может быть запущено **несколько X-серверов**, которые выводят графическую информацию на разные дисплеи с разными номерами: 0, 1, 3 и т.д.

Чтобы прикладные программы знали на каком компьютере и на каком дисплее находится X-сервер, используется переменная окружения **DISPLAY**, которая содержит информацию в формате:

<IP-адрес X-сервера>:<номер дисплея>.<номер экрана дисплея>

Таблица 2.2. Примеры задания значений переменной DISPLAY

Значение переменной DISPLAY	Номер порта соединения	Пояснение
:0	--	X-сервер на локальном компьютере, дисплей №0, экран №0
:1.0	--	X-сервер на локальном компьютере, дисплей №1, экран №0
asu.tusur.ru:4	6004	X-сервер на компьютере asu.tusur.ru, дисплей №4, экран №0, номер UDP-порта - 6004
192.168.1.17:2	6002	X-сервер на компьютере 192.168.1.17, дисплей №0, экран №0, номер UDP-порта - 6002

Многие прикладные программы имеют специальный параметр **-display**, после которого можно указать IP-адрес и номер дисплея вывода. Например, запуск программы **xterm** в виде:

xterm -display foo:1<Enter>

выведет консольную программу на дисплей №1 компьютера **foo**.

Если на компьютере запущено несколько X-серверов, то следует пользоваться «горячими клавишами»:

- **<Ctrl>+<Alt>+<Backspace>** - приведет к остановке X-сервера;
- **<Ctrl>+<Alt>+<F#>** - переключение между консолями Linux.

Замечание 11.

Запустить X-сервер может только пользователь **root**. Запуск производится с помощью программы **xinit**, общий формат запуска которой имеет вид:

xinit <клиент> -- <дисплей> <сервер>

где **<клиент>** - параметры, определяющие клиентскую часть;

<дисплей> - адрес компьютера и номер дисплея, где загрузится X-сервер;

<сервер> - серверная часть параметров.

Следует заметить, что структура запуска программы **xinit** — довольно сложна и требует высокой квалификации.

На практике используются два основных варианта запуска X-сервера:

- **Вариант 1.** X-сервер запускается менеджером дисплея, например, **gdm** (*gnome display manadger*), во время загрузки ОС в графическом режиме.
- **Вариант 2.** X-сервер запускается пользователем из текстового режима посредством скрипта **startx**.

В любом из используемых вариантов, конечным результатом этих усилий будет запуск программы **xinit** с нужными параметрами. Здесь следует учесть, что основными файлами, содержащими параметры запуска X-сервера, являются:

- **/etc/X11/Xsession;**
- **/etc/xinit/xinitrc;**
- **/etc/xinit/xserverrc.**

Замечание 12.

Если эти файлы скопировать в домашнюю директорию пользователя с именами **.Xsession**, **.xinitrc** и **.xserverrc**, а затем отредактировать нужным образом, то X-сервер будет запускаться с параметрами указанными в этих файлах.

Кроме перечисленных выше свойств, X-сервер имеет набор механизмов различного уровня защиты от несанкционированного доступа. Современные дистрибутивы Linux, такие как ОС Xubuntu, поставляются с параметрами, запрещающими X-серверу подключать программы удаленных компьютеров.

Замечание 13.

Поскольку подробное обсуждение средств защиты выходит за рамки данного пособия, студенту следует использовать интерактивное справочное руководство **man** с параметром **Xsecurity**.

В дистрибутивах **Linux Xubuntu** используется оконный менеджер **LightDM**. Чтобы разрешить подключение сети, необходимо:

- 1) открыть файл **/etc/lightdm/lightdm.conf**;
- 2) в секцию **[SeatDefaults]** добавить строку:

xserver-allow-tcp=true

- 3) создать еще одну секцию:

[XDMCPServer]
enabled=true

Перезапустить оконный менеджер командой:

sudo restart lightdm

Замечание 14.

В дистрибутиве ОС УПК АСУ, **все необходимые настройки X-сервера уже проведены.**

Изучив основные элементы архитектуры X Window System, перейдем к пошаговому изложению операций, которые студент должен выполнять на каждой лабораторной работе от момента запуска дистрибутива ОС УПК АСУ до момента запуска на кластере кафедры АСУ интегрированной среды разработки IDE Eclipse.

Шаг 1. Загрузить ОС УПК АСУ, в графическом режиме, и выполнить все необходимые действия по настройке сети компьютера согласно подразделам 2.1 и 2.2 данного руководства.

Шаг 2. С помощью команды *ifconfig*, определить и запомнить IP-адрес компьютера, на котором проводится лабораторная работа.

Шаг 3. В консоли, от имени пользователя **root**, следует выполнить команду:

xhost +cluster.asu.tusur.ru

которая разрешит пользователям компьютера **cluster.asu.tusur.ru** подключаться к вашему X-серверу.

Замечание 15.

Если в команде **xhost** заменить '+' на '-', то доступ с компьютера **cluster.asu.tusur.ru** к вашему X-серверу будет запрещен.

Соответственно, для проверки, что на компьютере разрешено подключение к X-серверу, используется команда:

xhost

Эта команда покажет список и уровень защищенности всех разрешенных соединений.

Шаг 4. Командой **ssh** подключиться к серверу **cluster.asu.tusur.ru**, используя правила, изложенные в разделе 2.2 данного пособия.

Шаг 5. Находясь на консоли удаленного сервера, отредактировать файл **.bashrc**, установив переменную **DISPLAY**, как показано на следующем рисунке:

Замечание 16.

В качестве параметра IP-адреса, студент должен указать **IP-адрес** своего компьютера, определенного ранее на **шаге 1**.

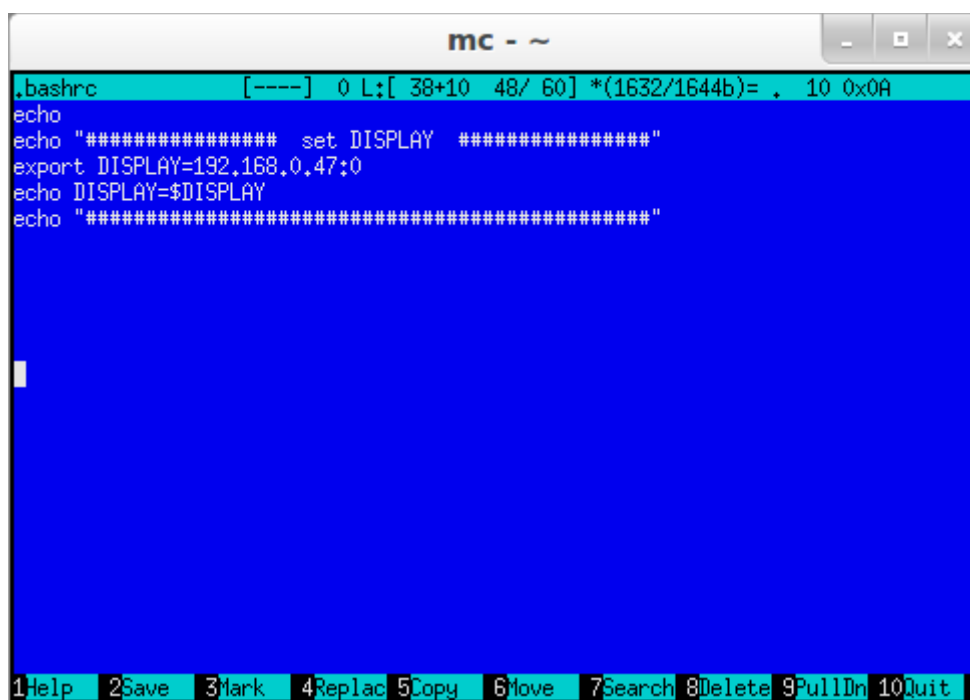


Рис. 2.6. Редактирование системной переменной DISPLAY

Переменную **DISPLAY** всегда нужно проверять, поскольку IP-адрес компьютера студента может изменяться.

Далее, следует выйти из редактора и перезапустить скрипт **.bashrc**, как показано на рисунке 2.7, чтобы убедиться в правильности установки переменной **DISPLAY**.

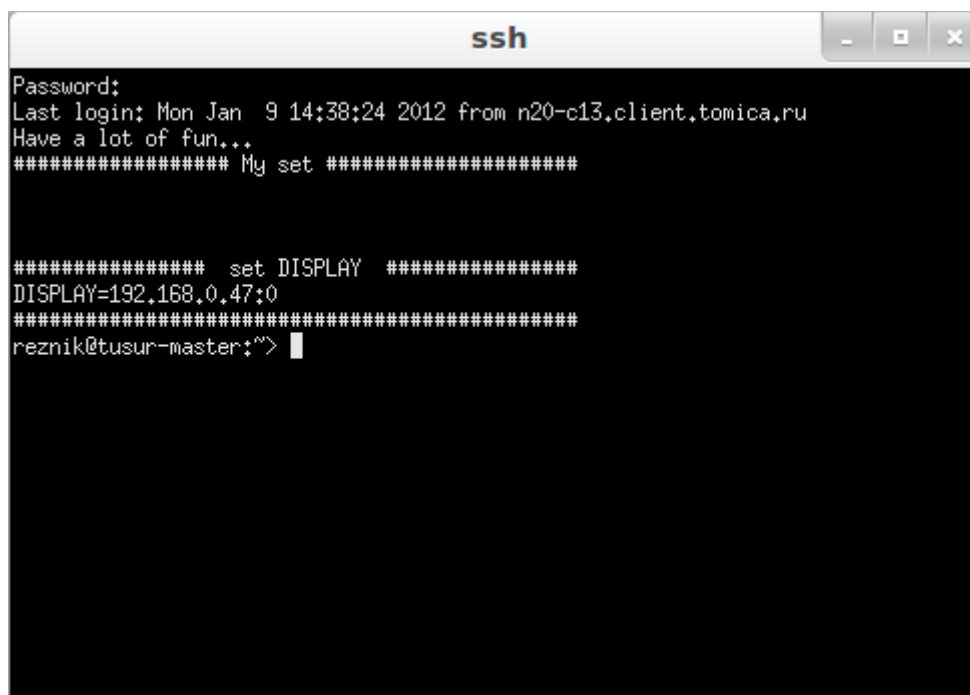


Рис. 2.7. Проверка установки переменной DISPLAY

Шаг 6. Нужно дополнительно убедиться в правильности настроек, запустив на удаленном компьютере команду **xterm**. При правильных настройках, которые были проведены на предыдущих шагах, на рабочем столе должно появиться новое окно консоли, озаглавленное **xterm**.

Выйти из этой консоли можно командой **exit**.

Замечание 17.

Все перечисленные **шесть шагов** следует выполнять перед самым **началом выполнения работ на кластере кафедры АСУ**.

3. КОНТРОЛЬ ВЫПОЛНЕНИЯ РАБОТ

Текущий и заключительный контроль выполнения лабораторных работ является обязательной составляющей процесса обучения.

Текущий контроль преподаватель проводит следующими способами:

- индивидуальная регистрация посещения каждой лабораторной работы каждым бакалавром;
- наблюдение общего хода выполнения отдельной лабораторной работы в течение отведенного для ее выполнения времени;
- просмотр индивидуальных отчетов магистрантом и регистрация объема выполненной работы;
- оценка результатов, выполненных каждым магистрантом работ, по двум контрольным точкам.

Заключительный контроль проводится общей оценкой результатов обучения по всей дисциплине в виде **зачета**.

Последующие лабораторные работы проводятся в двух режимах:

- **режиме локальной работы** на рабочей станции магистранта;
- **режиме удаленного доступа** на кластере ЭВМ кафедры АСУ.

Чтобы завершить контроль умения работы магистранта на кластере ЭВМ кафедры АСУ, следует выполнить простейший контрольный пример, описание которого приведено в подразделе 3.1.

3.1. Удаленная разработка приложений на базе IDE Eclipse

Учебное задание магистранту:

- подключиться к кластеру кафедры АСУ и запускает интегрированную среду разработки (IDE) EclipsePTP;
- создать типовой проект на языке C, который выводит на консоль строку сообщения: «**Hello, world!!!**».

Навыки этого подраздела обеспечат студента умениями, необходимыми для выполнения лабораторной работы №5.

Запустив дистрибутив ОС УПК АСУ, настроив и проверив ее, как описано в разделе 2 данного учебного пособия, студент запускает Midnight Commander и переходит к работе с интегрированной средой разработки IDE Eclipse.

Перемещаясь по дереву файловой системы, студент заходит в директорию **/home/asu/reznik/eclipsePTP/** и запускает на выполнение файл **eclipse**. В результате, на экране монитора появится окно **«Select a workspace»**, которое предлагает определить директорию рабочей области (см. рис. 3.1).

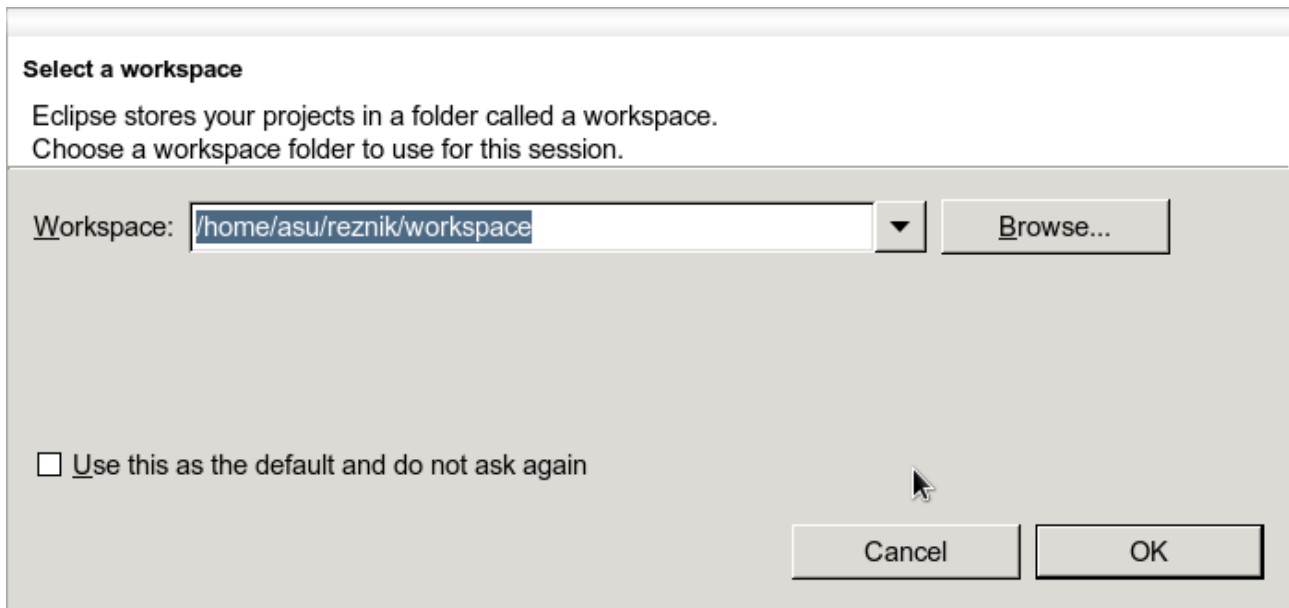


Рис. 3.1. Окно выбора директории проектов Eclipse

Магистрант редактирует строку ввода **«Workspace:»**, придерживаясь формата **<Домашняя директория студента>/workspace**. Затем воздействует на кнопку **ОК...** Загрузится среда разработки Eclipse.

Замечание 18.

Если рабочая директория среды разработки выбирается первый раз, то появится окно с предложением выбрать разделы инструментальной среды. Следует выбрать раздел **«Workbench»** и появится **окно среды разработки**. При последующих выборах этой директории, рабочая среда разработки Workbench будет показана сразу.

Ниже, на рисунках 3.2 и 3.3, показаны оба окна запуска среды EclipsePTP. Приступаем к созданию контрольного примера на языке C.

Интегрированная среда разработки IDE EclipsePTP, как и большинство других современных сред разработки, делит создаваемые программные продукты на проекты. **Все создаваемые проекты**, в пределах рабочей области Workspace, **должны иметь уникальные имена**.

С целью упорядочения процесса проведения занятий, а также последующей отчетности и контроля, студент на каждой лабораторной работе создает один проект. Соответственно, проекты именуются по порядку: lab1, lab2, lab3 и т.д.

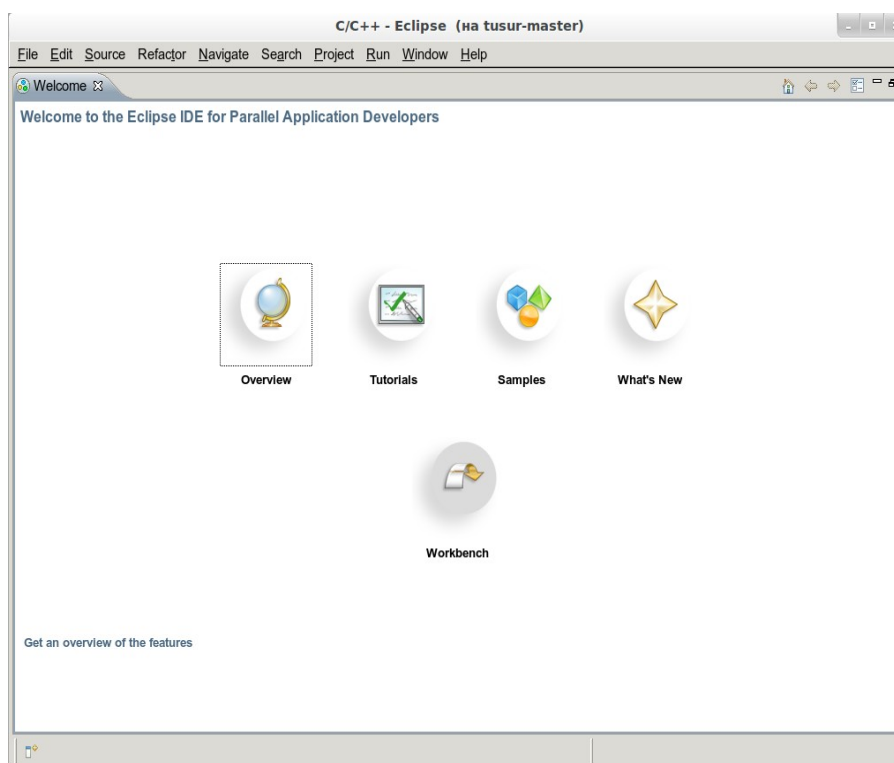


Рис. 3.2. Окно Eclipse при первом создании рабочей области

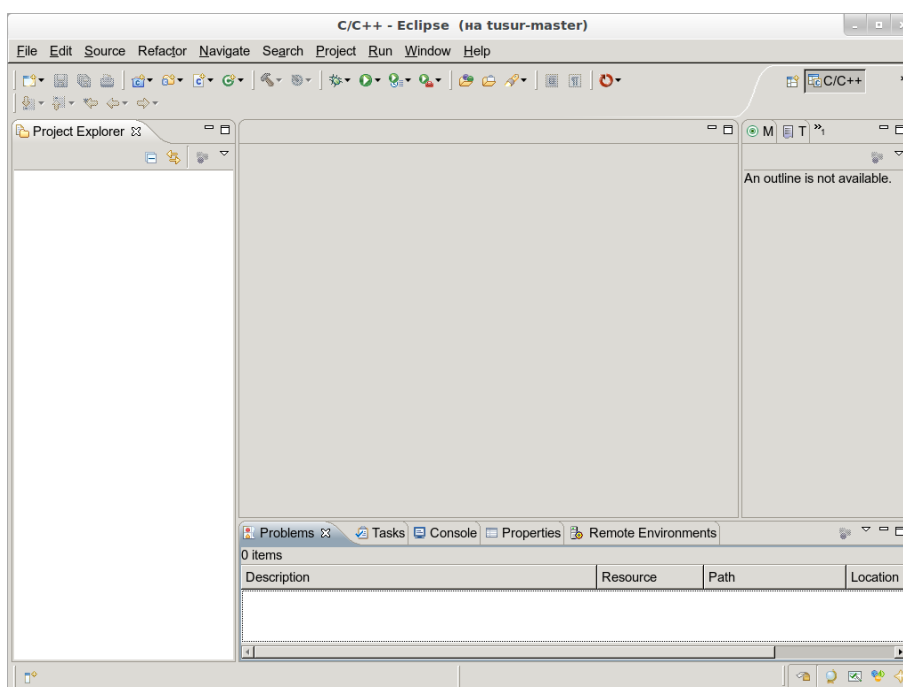


Рис. 3.3. Рабочее окно среды разработки Eclipse

Задание на разработку ПО

В данной лабораторной работе, студент должен создать проект с именем **lab1** для типовой программы на языке C. Эта программа должна выводить на стандартное устройство вывода (консоль) текст:

Hello, world!!! <Фамилия студента> <Имя студента>

Учебная цель данного задания:

- **освоение** и закрепление знаний работы с интегрированной IDE Eclipse;
- **уточнение** настроек рабочей среды разработки под локальную ОС, а также под системное и инструментальное ПО кластера ЭВМ кафедры АСУ;
- **демонстрация** студентом практических знаний по разработке ПО на языке С;
- **практическое закрепление** полного технологического цикла разработки ПО в пределах темы лабораторной работы №1.

Чтобы максимально упростить рутинные действия студента при запуске IDE Eclipse, следует дополнительно отредактировать файл **.bashrc**, как показано на рисунке 3.4. Тогда запуск Eclipse можно будет производить с консоли командой: **\$myIDE**

```
mc - ~
.bashrc [----] 0 L: [ 35+15 50/ 61] *(1687/1698b)= . 10 0x0A
echo
echo "##### set Eclipse #####"
export myIDE="/home/asu/reznik/eclipsePTP/eclipse -data /home/asu/reznik/workspa
echo myIDE=$myIDE
echo
echo "##### set DISPLAY #####"
export DISPLAY=192.168.0.47:0
echo DISPLAY=$DISPLAY
echo "#####
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 3.4. Редактирование переменной среды myIDE

Замечание 19.

Программа **eclipse**, в качестве значения ключа **-data** принимает значение директории, которая будет рассматриваться как **Workspace**.
Студент должен правильно установить значение этого параметра.

Поскольку IDE Eclipse — сложная модульная система, первоначально созданная для разработки ПО на языке Java, то основная масса литературы, посвященная Eclipse и примерам разработки на ней, содержит контекст для языка Java.

В последнее время, некогда единый проект интегрированной среды Eclipse, возглавляемый Eclipse Foundation, стал делиться на специализированные проекты, среди которых появилась среда разработки для ПО на C/C++ (среда CDT — C Development Tool).

Позже появился еще один специализированный проект, поддерживающий разработку приложений с параллельными вычислениями (Eclipse IDE for Parallel Application Developers). Другое название этого проекта — **Eclipse PTP (Eclipse Parallel Tools Platform)**.

На кластере ЭВМ кафедры АСУ установлена среда разработки Eclipse PTP версии 3.7 (проект Indigo). Эта среда обеспечивает разработку ПО на C, C++ и Fortran.

С целью повышения теоретического уровня подготовки по технологии работы с интегрированной средой разработки Eclipse, студенту рекомендуется:

- **изучить** архитектуру Eclipse в пределах статьи [2];
- **ознакомиться** и использовать как справочное руководство по среде Workbench русскоязычный интернет-ресурс [3];
- **использовать** как официальное справочное руководство по всем технологическим аспектам Eclipse англоязычный интернет-ресурс [4].

3.2. Контроль выполнения лабораторной работы №1

За время обучения по дисциплине «Архитектура вычислительных комплексов», магистрант должен подготовить единый отчет.

Шаблон отчета, из файла *c:\asu14.04upk\avk-010400.68\lab1\avk-report.doc*, следует скопировать на личный flashUSB, отредактировать титульную страницу и регулярно заполнять в процессе выполнения всех лабораторных работ по данной дисциплине.

В целом, текст отчета заполняется **в произвольной форме**, выбранной самим обучающимся.

Обязательным является:

- деление отчета на разделы, соответствующие отдельной работе;
- отчет должен иметь «СОДЕРЖАНИЕ» с перечисленными разделами и указанием страниц на каждый раздел;
- разделы должны быть пронумерованы; номер раздела должен быть равен номеру соответствующей лабораторной работы;

- допускается, по желанию магистранта, разбивать разделы на подразделы; в таком случае, эти подразделы должны быть отражены на странице «СОДЕРЖАНИЕ»;
- магистрант несет полную ответственность за сохранность отчета.

По требованию преподавателя, отчет должен быть предоставлен для проверки и, при необходимости, скопирован в указанное преподавателем место.

Лабораторная работа №1 **считается выполненной**, когда магистрант способен уверенно продемонстрировать:

- запуск ОС УПК АСУ с личного flashUSB и настройку параметров сети для выполнения лабораторных работ;
- защищенное подключение к кластеру ЭВМ кафедры АСУ;
- запуск Eclipse и демонстрацию запуска программ созданных проектов.

ЛИТЕРАТУРА

1. Костромин В. Графический интерфейс Linux: Интернет-ресурс - <http://rus-linux.net/papers/xwin/X-Window.html>.
2. Рахимбердиев А. Проект Eclipse: Интернет-ресурс - <http://www.rsdn.ru/article/devtools/eclipse.xml>.
3. Eclipse Workbench. Основной учебник: Интернет ресурс — <http://khpi-iip.mipk.kharkiv.edu/library/extent/prog/eclWB/1.html>.
4. Help-Eclipse Platform: Интернет-ресурс - <http://help.eclipse.org/indigo/index.jsp>.

Учебное издание

Резник Виталий Григорьевич

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ

Методические рекомендации для выполнения лабораторной работы №1
«Подготовка работы на УПК АСУ» по дисциплине «Архитектура
вычислительных комплексов» для студентов уровня основной образовательной
программы магистратура
направления подготовки 010400.68 «Прикладная математика и информатика»
профиля «Математическое и программное обеспечение вычислительных
комплексов и компьютерных сетей».

Учебно-методическое пособие

Усл. печ. л. . Тираж 100. Заказ .

Томский государственный университет
систем управления и радиоэлектроники
634050, г. Томск, пр. Ленина, 40