

The UNIX System



- Ø Ядро – Сърцето на операционната с-ма
 - Ø Process scheduling
 - Ø I/O control (accesses)
- Ø Обвивка (Shell) – посредник между потребителя и ядрото
- Ø Инструменти и приложения
 - Ø Достъпни от обвивката
 - Ø Могат да бъдат изпълнявани независимо от обвивката

UNIX системно програмиране

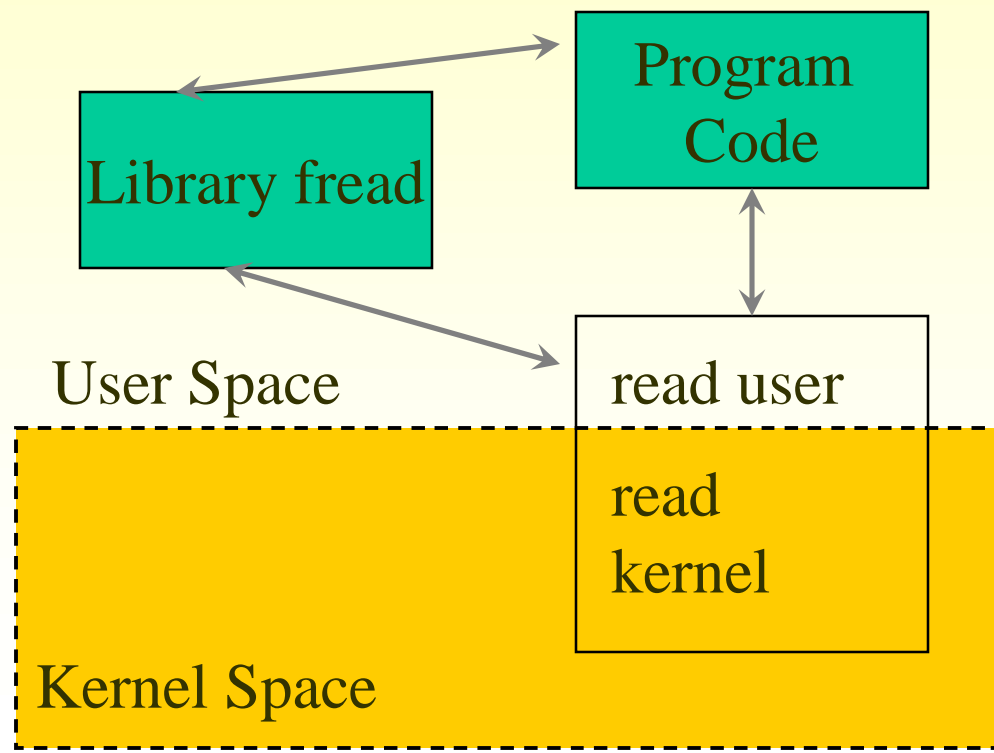


- Ø Програмите извършват *системно извикване*
- Ø Системното извикване е процедура, която се изпълнява от операционната с-ма
- Ø Типове системни извиквания
 - Ø Файл I/O
 - Ø Управление на процеси (Process management)
 - Ø Комуникация между процеси “Inter-process communication” (IPC)
 - Ø Обработка на сигнали (Signal handling)



Ø Системни повиквания

Ø Интерфейс към ядрото



Основни файлови I/O



- Ø Процесите пазят списък на отворените файлове
- Ø Файловете могат да се отварят за четене и запис
- Ø Всеки файл се указва от дескриптор на файла *file descriptor* (integer)
- Ø Три файла се отварят автоматично:
 - Ø **FD 0: стандартен вход**
 - Ø **FD 1: стандартен изход**
 - Ø **FD 2: стандартна грешка**

Файлови I/O системни извиквания: `open()`



Ø `fd = open(path, flags, mode)`

Ø път (`path`): символен низ, абсолютен или относителен път

Ø флагове (`flags`):

Ø `O_RDONLY` – отваряне за четене

Ø `O_WRONLY` – отваряне за писане

Ø `O_RDWR` – отваряне за четене и писане

Ø `O_CREAT` – създаване на файл, ако не съществува

Ø `O_TRUNC` – презапис на файла, ако съществува

Ø `O_APPEND` – пиши само в края на файла

Ø режим (`mode`): указване на права, ако се използва `O_CREAT`

Файлово I/O системно повикване: `close()`

Ø `retval = close(fd)`

Ø Затваряне на отворен файл или дескриптор

Ø Връща 0 при успех и -1 при грешка

Файлово I/O системно повикване: `read()`



Ø `bytes_read = read(fd, buffer, count)`

Ø Чете до `count` байта от файла и ги слага в буфер `buffer`

Ø `fd`: файлов дескриптор

Ø `buffer`: указател към масив

Ø `count`: брой на байтовете, които трябва да се прочетат

Ø Връща броя на прочетените байтове или `-1` при грешка

Файлово I/O системно извикване: `write()`



Ø `bytes_written = write(fd, buffer, count)`

Ø Записва `count` на брой байта от буфера `buffer` във файла

Ø `fd`: дескриптор на файла

Ø `buffer`: указател към масив

Ø `count`: брой байтове за запис

Ø Връща броя на записани байтове или `-1` при грешка

Системно извикване: `lseek()`



`Ø retval = lseek(fd, offset, whence)`

Ø Премества указателя на файла в ново положение

Ø `fd`: дескриптор на файла

Ø `offset`: отместване в брой байтове

Ø `whence`:

Ø `SEEK_SET` — отместване от началото на файла

Ø `SEEK_CUR` — отместване от сегашната позиция

Ø `SEEK_END` — отместване от края на файла

Ø Връща отместването от началото на файла или `-1`

UNIX File access primitives



- Ø open – отваря за четене писане или създава празен файл
- Ø creat – създава празен файл
- Ø close – затваря файл
- Ø read – чете от файл
- Ø write – пише във файл
- Ø lseek – ходи на определен байт във файла
- Ø unlink – трие файла
- Ø remove – трие файла
- Ø fcntl – контролира атрибутите свързани с файла

File I/O using FILEs



Ø Повечето UNIX програми ползват I/O функции
от ВИСОКО НИВО

`Øfopen()`

`Øfclose()`

`Øfread()`

`Øfwrite()`

`Øfseek()`

Ø Те ползват **FILE** типът данни (datatype) вместо
дескрипторите на файлове

Ø Необходимо е да се включи **stdio.h**

Използване на типове данни с файлове I/O



- Ø Всички функции до сега използваха първични байтове за файлове I/O, но данните за програмите са в някакъв смислен тип (**int, char, float, etc.**)
- Ø **fprintf(), fputs(), fputc()** – се използват за писане във файл
- Ø **fscanf(), fgets(), fgetc()** – се използват за четене от файл

Програмиране в UNIX среда

Графична среда.

Потребителски интерфейс



Ø Цел: да осигури възможност за взаимодействие с компютъра, както и с устройствата, които той контролира

Ø Конзолен и графичен

Ø Графичен интерфейс – често се използва като аналог на

Ø Прозорци – съдържа информацията

Ø Икони – представяне на прозорците

Ø Менюта – групи от команди

Ø Показалец

Минимизира се нуждата крайният потребител да е експерт по компютри при извършване на своята работа

Ø Този тип взаимодействие с компютъра е разработен от XEROX (1981 г.) и популяризиран от Macintosh компютрите (1984 г.)

XEROX STAR - 1981



XEROX 6085 Workstation

User-Interface Design

To make it easy to compose text and graphics, to do electronic filing, printing, and mailing all at the same workstation, requires a revolutionary user interface design.

Fit-map display - Each of the pixels on the 19" screen is mapped to a bit in memory; thus, arbitrarily complex images can be displayed. The 6085 displays all facts and graphics as they will be printed. In addition, familiar office objects such as documents, folders, file drawers and in-baskets are portrayed as recognizable images.

The mouse - A unique pointing device that allows the user to quickly select any text, graphic or office object on the display.

See and Point

All functions are visible to the user on the keyboard or on the screen. The user does filing and retrieval by selecting them with the mouse and touching the MOVE, COPY, DELETE or PROPERTIES command keys. Text and graphics are edited with the TABUL keys.

Shorter Production Time:

Experience at Xerox with prototype work stations has shown shorter production times and thus lower costs, as a function of the percentage of use of the workstation. The following equation can be used to express this:

$$C = \frac{A + B}{1 - P}$$

where C is the cost per copy, A is the cost of the workstation, B is the cost of the copy, and P is the percentage of use of the workstation.

YEAR	Non 6085	6085
1978	95.2	45.8
1979	41.1	59.3
1980	45	55
1984	30	70
1986	10	90
1988	5	95

Table 1: Percentages of use of methods.

Activity under the old and the new methods

Figure 1: Data from Table 1 drive

Workstation usage percentages Table 1 and illustrated in Figure 1 and illustrated in Figure 1. 6085 users are likely to do the composition and layout, create process including printing and distribution.

Text and Graphics

To replace typesetting, the 6085 offers a choice of type fonts and sizes from 6 point to 36 point.

Here is a test piece of 18 point text.

Here is a test piece of 24 point text.

18-point text.

24-point text.

36-point text.

ГПИ: Общи характеристики



Ø Управление на прозорците + потребителска среда

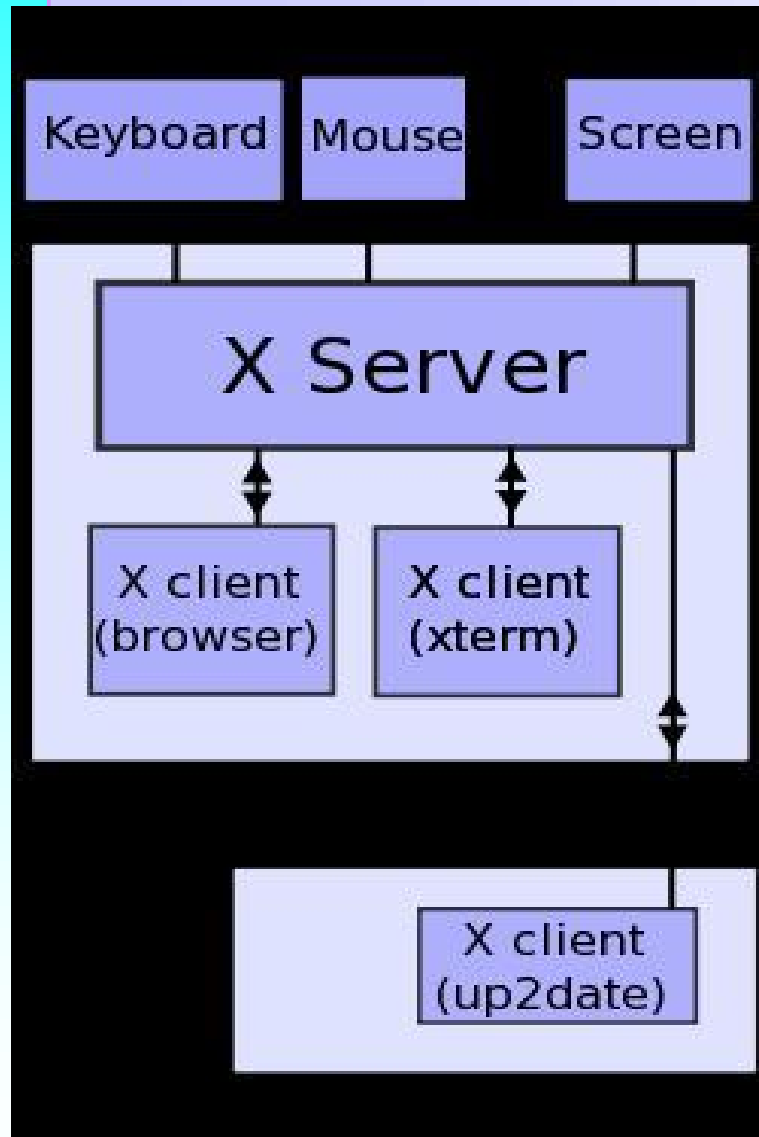
- Ø Предоставя възможностите, разработени от XEROX и Apple
- Ø Програми за:
 - Ø Настройка на системата (администрация)
 - Ø Офис приложения, текстообработка
 - Ø Интернет приложения
 - Ø Мултимедиини приложения
- Ø Обикновено програмите, написани за една потребителска среда, могат да се използват в друга, при наличието на необходимите библиотеки
- Ø Най-често използваните потребителски среди са GNOME и KDE

X прозоречна среда



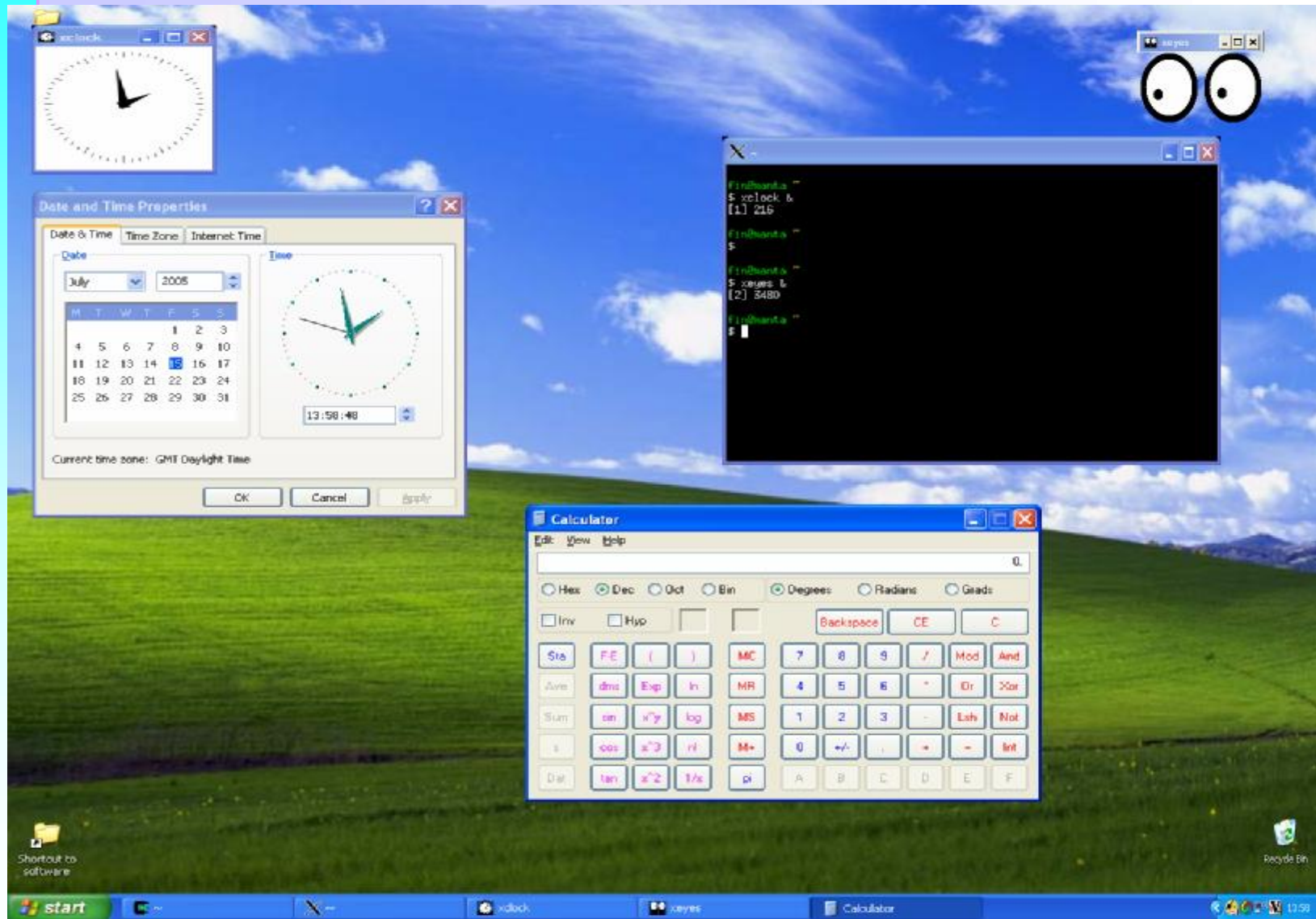
- Ø Започва да се разработва през 1984 и дава основната среда за конструиране на приложения за графичен интерфейс
 - Ø Не е част от операционната система, а представлява отделна обвивка от потребителски приложения, надстройка над ядрото
 - Ø Специално разработена за използване в мрежа – X – протокол за графични изображения на базата клиент-сървър
 - Ø Мрежово прозрачен протокол: не зависи от архитектурата на ОС
позволява оптималното използване на нехомогенни системи
- Ø X сървърът приема заявки за графичен изход от различни приложения и изпраща обратно потребителските действия (с мишка, клавиатура)
 - Ø Локалният екран служи като сървър на програмите, които изпълняваме на локалния или отдалечени компютри

X прозоречна среда



- Ø X не съдържа никакви спецификации за изгледа на прозорците, менюта и т.н.
 - Ø Позволява произволен изглед на потребителската среда
- Ø Почти винаги X се използва с цялостна потребителска среда като GNOME и KDE, а X е отговорен за взаимодействието между мишката и клавиатурата и екрана.
- Ø Съществуват две основни реализации на X
 - Ø XFree86
 - Ø X.org

X прозоречна среда за Windows



Л. Литов

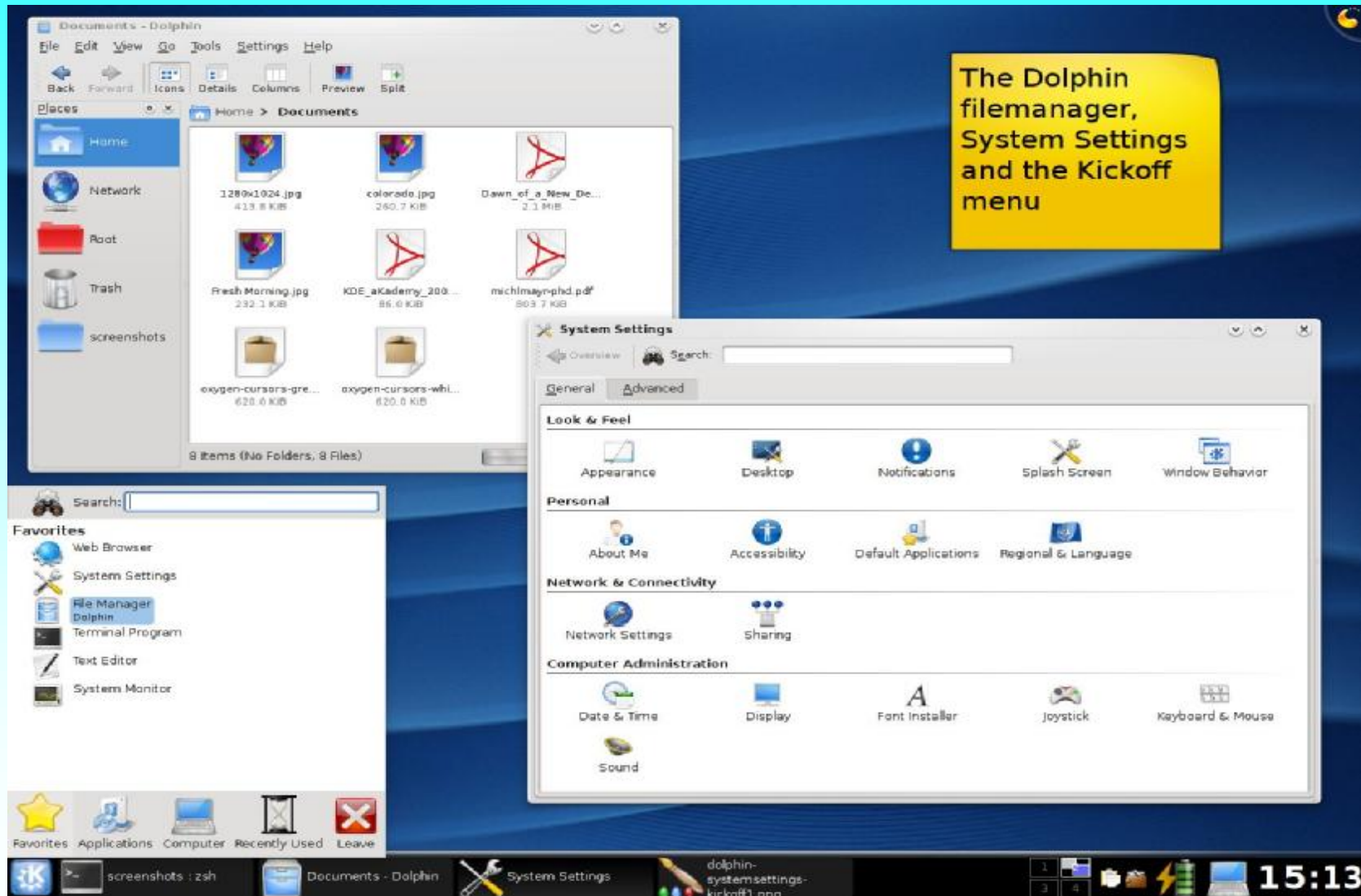
Програмиране в UNIX среда

KDE



- Ø Започва развитието си през 1996 г.
 - Ø Преминаването от набор от приложения към създаването на единна потребителска среда, в която може да се очаква предвидимо и съгласувано поведение на приложенията
 - Ø Името идва като наследник на CDE (common desktop environment)
- Ø Използва Qt библиотеката за създаване на графични приложения
 - Ø Не се разпространява под свободен лиценз през 1997
 - Ø Qt 4 (и KDE 4) са под GPL лиценз
- Ø KDE е достъпно навсякъде, където може да се инсталира Qt: UNIX, Linux, Mac OS X
- Ø Лесни за използване приложения, с множество възможности
- Ø Множество възможности за промяна на изгледа на потребителския интерфейс

KDE 4



Л. Литов

Програмиране в UNIX среда

GNOME



- Ø Проект, стартиран през 1997 г.
 - Ø Цел: създаването на изцяло свободна потребителска среда (свободна за разпространение и промяна, за разлика от KDE по това време)
 - Ø Простота
 - Ø Възможност от използването на среда от всеки на родния му език
- Ø Приложения, които имат само нужните възможности
 - Ø Бързина
 - Ø По-малко бъгове
- Ø Възможност за свободен избор на език за програмиране на приложения, интегриращи се със средата: C++ (gtkmm), Java (java-gnome), Ruby (ruby-gnome2), C#, (Gtk#), Python (PyGTK), Perl (gtk2-perl)

GNOME



The screenshot displays the GNOME desktop environment. The desktop background is a blue sky with clouds. The top panel shows the system menu with icons for Programs, Places, System, and various system utilities. The taskbar at the bottom contains icons for the Friends list, gbgoffice, the VENI file browser, and the Terminal.

The main window is the 'VENI - файлов браузер' (VENI - file browser) showing a directory tree with folders like 'Домашна папка', 'Файлова система', and 'VENI'. The 'VENI' folder is expanded, showing subfolders like 'ACER-backup', 'Docs', 'mp3', 'pics', 'programs', 'recycled', 'System Volume Information', 'tmp', 'video', and 'x'. A context menu is open over the 'transfusion' file, listing related terms such as 'transform', 'transformable', 'transformation', 'transformer', 'transfuse', 'transgression', and 'transgress'.

The 'Терминал' (Terminal) window shows the command prompt with the current directory set to '/data/veni':
[rcab123]:/data/veni>

The 'Списък с приятели' (Friends list) window shows a list of contacts under the 'MY' and 'Friends' categories, including 'veni', 'Adi', 'Chaves', 'Desi', and 'Feb(MC)**'.

The 'gbgoffice' window shows a dictionary entry for 'transfusion' with its phonetic transcription [trɪnsˈfjuːʃən] and definitions: 'п преливане (и на кръв), трансфузия' and 'blood TRANSFUSION п преливане на кръв'.

Л. Литов

Програмиране в UNIX среда

Приложения под GNOME/ KDE



	<u>GNOME</u>	<u>KDE</u>
Управление на прозорци	<i>Metacity</i>	<i>Kwin</i>
Управление на файлове	<i>Nautilus</i>	<i>Dolphin</i>
Базова библиотека	<i>GTK+</i>	<i>Qt</i>
Текстов редактор	<i>Gedit</i>	<i>Kate</i>
Слушане на музика	<i>Rhythmbox</i>	<i>Amarok</i>
Гледане на филми	<i>Totem</i>	<i>Kaffeine</i>
Записване на файлове	<i>Brasero</i>	<i>K3b</i>
Разглеждане на картинки	<i>Eye of GNOME</i>	<i>Kview</i>
Офис приложение	<i>GNOME Office</i>	<i>Koffice</i>
Електронна поща	<i>Evolution</i>	<i>Kontact</i>
Клиент за съобщения	<i>Pidgin</i>	<i>Kopete</i>
PS / PDF документи	<i>Evince</i>	<i>Okular</i>

Възможност за избор – множество алтернативни приложения

3D потребителска среда



Л. Литов

Програмиране в UNIX среда

Литература:



- Ø <http://www.wylug.org.uk/talks/2003/04/unix.pdf>
- Ø <http://ce.sharif.edu/courses/ssc/unix/resources/root/Slides/unixhistory.pdf>
- Ø <http://www.cs.uga.edu/~eileen/1730/Notes/intro-UNIX.ppt>
- Ø <http://remus.rutgers.edu/cs416/F01>
- Ø <http://www.cs.virginia.edu/~cs458/>
- Ø <http://www.bobbooth.staff.shef.ac.uk/hpcs/materials/material.html>
- Ø <http://www.comm.utoronto.ca/~jorg/teaching/ece461>
- Ø <http://home.iitk.ac.in/~navi/sidbilinuxcourse/>
- Ø <http://www.cs.washington.edu/homes/bershad/Mac/ssh/practicalmagic.pdf>
- Ø <http://www.cs.cf.ac.uk/Dave/C/CE.html>
- Ø <http://www.le.ac.uk/cc/tutorials/c/ccccintr.html>
- Ø <http://www.shef.ac.uk/uni/academic/N-Q/phys/teaching/phy225/index.html>