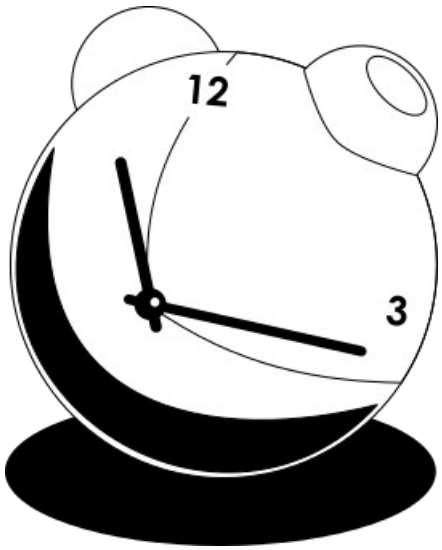


John Brown

FreeBSD 12.1

Десктоп за 20 минут

John Brown. FreeBSD 12.1. Десктоп за 20 минут.

Статья посвящена построению десктопа на основе операционной системы FreeBSD 12.1 и поможет начинающим пользователям настроить графическое окружение. Цель статьи — описать способ быстрой настройки системы, который сводится к выполнению практически одной команды. При этом в статье также уделено внимание и тому, как сделать настройку вручную. При использовании описанного в статье флеш-накопителя, можно развернуть десктоп менее чем за 20 минут, при этом не имея доступа к сети.

Оглавление

Установка FreeBSD....5
Редактор vi....13
Настройка консоли....19
Драйвер консоли newcons....19
Выбор раскладки клавиатуры....20
Настройка login-класса....20
Подготовка к установке дополнительных программ....20
Получение и обновление дерева портов....21
Установка пакетного менеджера....21
Добавление репозиториев....21
Удобная работа с флеш-накопителем....21
Настройка....21
Как пользоваться?....22
Как определить серийный номер флеш-накопителя?....23
Конструирование десктопа....24
Настройка графического интерфейса....24
Xorg....24
Установка....24
Настройка....24
Отображение шрифтов на экране....25
Шрифты в приложениях....26
Оконный менеджер Fluxbox....27
Установка....27
Настройка FluxBox....27
Настройка xxkb....27
Запуск....28
Графическое приглашение на вход в систему....28
Установка....28
Настройка....29
Изменение разрешения экрана и частоты обновления....30
Внешний вид программы Xterm....30
Установка приложений....30
Загрузочный флеш-накопитель....32
Персональный репозиторий на флеш-накопителе....32
Подготовка....32
Загрузка пакетов из официального репозитория....34
Использование персонального репозитория....35
Синхронизация персонального репозитория с официальным....35
Преимущество персонального репозитория....35

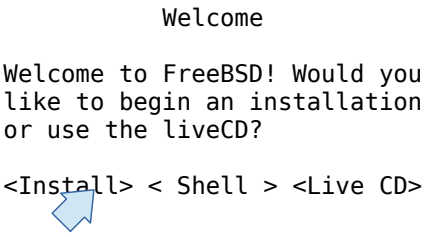
Создание пакета для приложения, собранного из порта....36
Пример создания пакета для fluxbox....36
Как собрать дескоп за 20 минут....37
Шаги для сборки десктопа....37
Как можно разнообразить десктоп....39
Часы со стрелками....39
Календарь....40
Приложение....41
Скрипт mpi....41
Скрипт fbMenuModify....45
Скрипт miniRepo....47
Скрипт usbdrive....50
Скрипт usb_mount_clean.sh....52
Скрипт wpwc....52

Установка FreeBSD

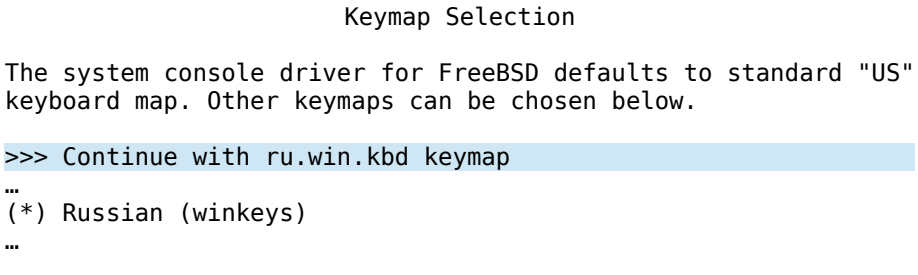
Внимание! При выполнении указанных далее действий, вся имеющаяся информация на жестком диске будет удалена.

FreeBSD Installer

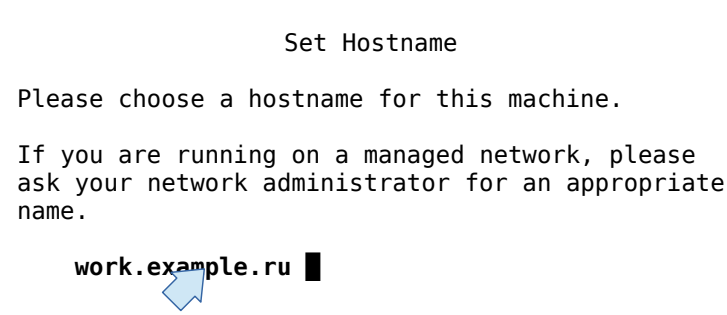
1



2



3



4

```
                        Distribution Select

Choose optional system components to install:

[ ] base-dbg    Base system (Debugging)
[ ] doc         Additional Documentation
[ ] kernel-dbg  Kernel (Debugging)
[ ] lib32-dbg   32-bit compatibility libraries (Debugging)
[*] lib32       32-bit compatibility libraries
[ ] ports       Ports tree
[ ] src         System source code
[ ] tests       Test suite
```

5

```
                            Partitioning

How would you like to partition your disk?

  Auto (UFS)    Guided Disk Setup
  Manual        Manual Disk Setup (experts)
  Shell         Open a shell and partition by hand
  Auto (ZFS)    Guided Root-on-ZFS
```

6

```
                            Partitioning

Select the disk on which to install FreeBSD

 ada0   466GB   ATA Hard Disk <ST3500413AS>
…
```

7

```
                            Partition

Would you like to use this entire disk
(ada0) for FreeBSD or partition it to
share it with other operating systems?
Using the entire disk will erase any
data currently stored there.

      <Entire Disk>   <Partition>
```

8

```
                            Confirmation

This will erase the disk.
Are you sure you want to
proceed?

                <Yes>   <No>
```

9

```
                        Partition Scheme

              Select a partition scheme for this
              volume:

               APM    Apple Partition Map
               BSD    BSD Label
               GPT    GUID Partition Table
               MBR    DOS Partition
               PC98   NEC PC9801 Partition Table
               VTOC8  Sun VTOC8 Partition Table
```

10

```
                        Partition Editor

            Please review the disk setup. When complete, press
            the Finish button.

             ada0             466GB       GPT
               ada0p1         512KB       freebsd-boot
               ada0p2         462GB       freebsd-ufs     /
               ada0p3         3,8GB       freebsd-swap    none

            <Create> <Delete> <Modify> <Revert> <Auto> <Finish>
```

11

```
                          Confirmation

              Your change will now be written to disk. If you
              have chosen to overwrite existing data, it will
              be PERMANENTLY ERASED. Are you sure you want to
              commit you change?

                    <Commit>   <Revert & Exit>   <Back>
```

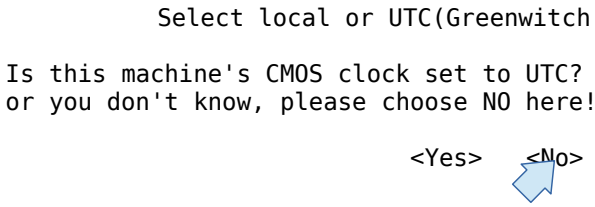
12

```
Please select a password for the system management account (root):
Typed characters will not be visible.
Changing local password for root

New Password: ********
Retype New Password: ********
```

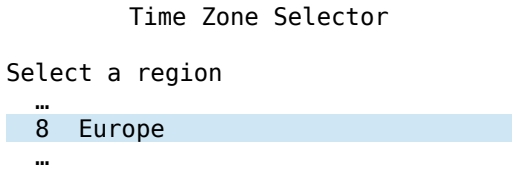
13

```
                        Network Configuration

              Please select a network interface to configure:

              …
                              <Ok>   <Cancel>
```

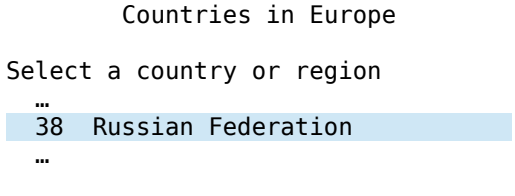
14



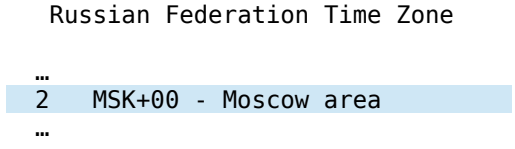
15



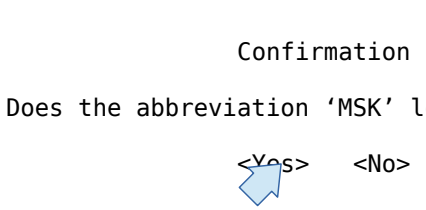
16



17



18



19

20

```
                          System Configuration

          Choose the services you would like to be started at boot:

          [ ] local_unbound  Local caching validating resolver
          [*] sshd           Secure shell daemon
          [*] moused         PS/2 mouse pointer on console
          [ ] ntpd           Synchronize system and network time
          [ ] powerd         Adjust CPU frequency dinamically if supported
          [ ] dumpdev        Enable kernel crash dumps to /var/crash
```

21

```
                              System Hardening

                       Choose system security hardening options:

    [ ]  0 hide_uids        Hide processes running as other users
    [ ]  1 hide_gids        Hide processes running as other groups
    [ ]  2 hide_jail        Hide processes running in jails
    [ ]  3 read_msgbuf      Disable reading kernel message buffer for unprivileged users
    [ ]  4 proc_debug       Disable process debugging facilities for unprivileged users
    [ ]  5 random_pid       Randomize the PID of newly created processes
    [ ]  6 clear_tmp        Clean the /tmp filesystem on system startup
    [ ]  7 disable_syslogd  Disable opening Syslogd network socket (disables remote logging)
    [ ]  8 disable_sendmail Disable Sendmail service
    [ ]  9 secure_console   Enable console password prompt
    [ ] 10 disable_ddtrace  Disallow DTrace destructive-mode
```

22

```
                          Add User Accounts

                      Would you like to add
                      users to the installed
                      system now?

                           <Yes>   <No>
```

23

```
Add Users:

Username: John
Full Name: John Smith
Uid (Leave empty for defaults):
Login group [John]:
Login group is John. Invite John into other groups?[ ]: wheel
Login class [default]: russian
Shell (sh csh tcsh nologin)[sh]: tcsh
Home directory [/home/John]:
Home directory permissions (Leave empty for default):
Use password based authentication?[yes]:
Use a empty password?(yes/no)[no]:
Use a random password?(yes/no)[no]:
Enter password: ********
Enter password again: ********
Lock out the account after creation?[no]:
…
OK?(yes/no): yes
…
Add another user?(yes/no): no
```

24

```
                          Final Configuration

Setup of your FreeBSD system is nearly complete. You can now modify
your configuration choices. After this screen, you will have an
opportunity to make more complex changes using a shell.

  Exit               Apply configuration and exit installer
  Add User           Add a user to the system
  Root Password      Change root password
  Hostname           Set system hostname
  Network            Networking configuration
  Services           Set daemons to run on startup
  System Hardening   Set security options
  Time Zone          Set system timezone
  Handbook           Install FreeBSD Handbook (requires network)
```

25

```
                          Manual Configuration

                The installation is now finished.
                Before exiting the installer, would
                you like to open a shell in the new
                system to make any final manual
                modification?

                          <Yes>    <No>
```

26

```
                    Complete

          Installation of FreeBSD
          complete! Would you like
          to reboot into the
          installed system now?

             <Reboot>   <LiveCD>
```

После перезагрузки получаем приглашение на вход в систему:

```
FreeBSD/amd64 (work.example.ru) (ttyv0)

login: John
Password: ********
```

Набираем имя и пароль и, если все верно, то видим справку и получаем возможность вводить команды:

```
FreeBSD 12.1-RELEASE r354233 GENERIC

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:   https://www.FreeBSD.org/security/
FreeBSD Handbook:      https://www.FreeBSD.org/handbook/
FreeBSD FAQ:           https://www.FreeBSD.org/faq/
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:        https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

Edit /etc/motd to change this login announcement.
FreeBSD's top(1) utility displays CPU statistics by default.
To display I/O activity for each process instead, run top like this:

top -m io

                -- Benedict Reuschling <bcr@FreeBSD.org>

John@work:~% 
```

Чтобы выключить компьютер, вводим команду:

```
John@work:~% shutdown -p now
```

Команда не выполняется — недостаточно прав. Поэтому становимся суперпользователем (потребуется пароль пользователя root)

```
John@work:~% su
Password: ********
```

Теперь команда выполнится успешно.

```
root@work:/home/John # shutdown -p now
```

Примечание
Нужно обратить внимание на приглашение командной оболочки, поскольку оно указывает от какого пользователя выполняется команда. При выполнении команд, приведенных в статье, это надо учитывать и при необходимости использовать команду su.

Пользователь может выключить компьютер, если он входит в группу operator. Добавить пользователя в группу operator можно следующей командой:

```
root@work:/home/John # pw groupmod operator -m John
```

Редактор vi

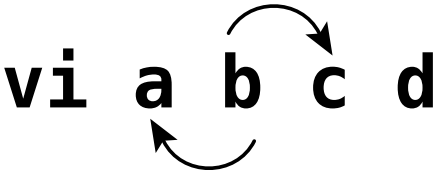
Поскольку параметры системы хранятся в текстовых файлах, то для дальнейшей настройки потребуется изучить какой-нибудь консольный текстовый редактор, например vi. Редактор vi не имеет «интуитивно понятного интерфейса» и не совсем прост в изучении, но зато он доступен во FreeBSD сразу после установки системы. Для работы с ним, на первых порах, понадобится небольшая памятка. Редактор vi имеет *режим ввода символов* и *режим ввода команд*. В режим ввода символов редактор переключается клавишами, указанными в памятке («Режим ввода символов / Коррекция при вводе»). Выход из этого режима (вход в режим ввода команд) — клавиша **Esc** (для надежности можно нажимать несколько раз).

Для заметок

Запустить / Открыть / Сохранить / Выйти

:n перейти к редактированию следующего
:next файла из списка аргументов

:N редактирование следующего файла
:Next из списка аргументов в новой области (окне)
(Ctrl+w - переход между окнами)



:p перейти к редактированию предыдущего
:prev файла из списка аргументов

:Prev редактирование предыдущего файла
из списка аргументов в новой области (окне)
(Ctrl+w - переход между окнами)

:q выйти
:q! выйти без сохранения

:args показать список аргументов
:ar Пример: [a] b c d

:rewind редактировать список с начала
:rewind! не сохранять изменения,
редактировать список с начала

:n A B C D загрузка файлов A B C D
для редактирования (новый список)

:e A загрузка файла A
:edit A для редактирования
(в список аргументов
он не попадает)

:Edit A тоже самое, только в отдельной
области (окне)
(Ctrl+w - переход между окнами)

:e# вернуться к редактированию
предыдущего файла

:w сохранить
:w newName сохранить как ...

:x сохранить и выйти
:wq

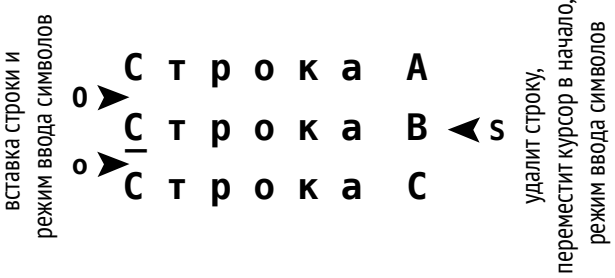
Режим ввода символов / Коррекция при вводе

переход в режим
ввода символов
(вставка)

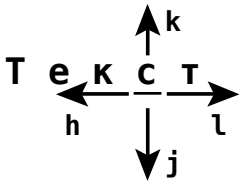
переход в режим
ввода символов
(вставка)

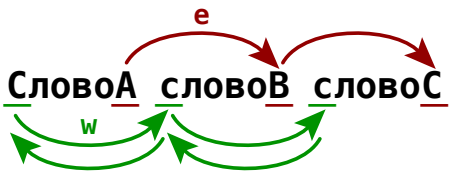
Ctrl+w
перепечатка слова в режиме ввода

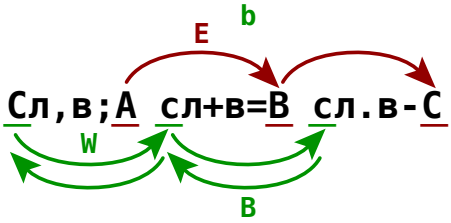
Ctrl+h
перепечатка символа в режиме ввода



Перемещение курсора по тексту







1G	Начало текста
G	Конец текста

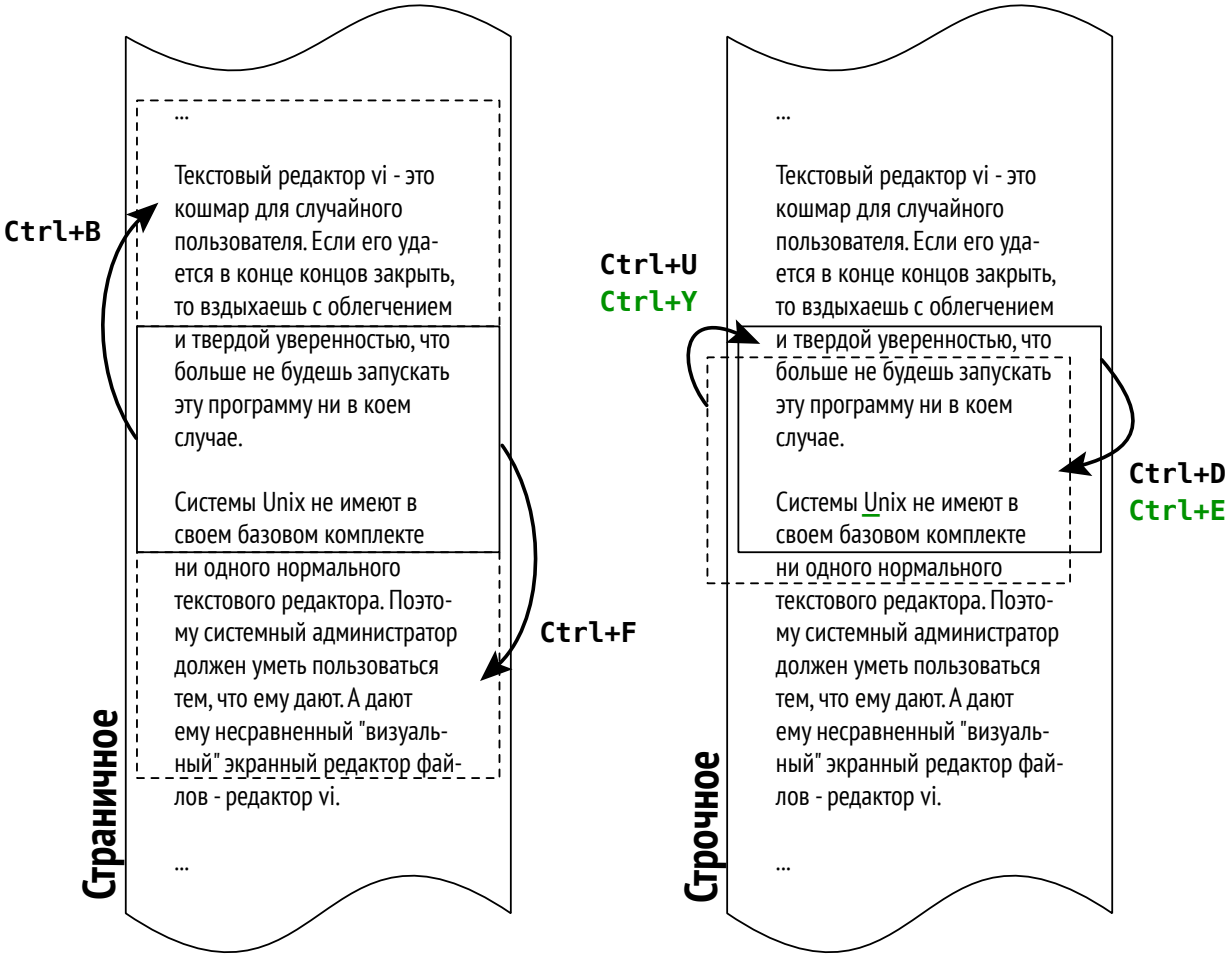
Это предложение? Да, это предложение! Потому что за знаком вопроса (?) идет два пробела.

А это предложение начинает новый параграф или абзац.

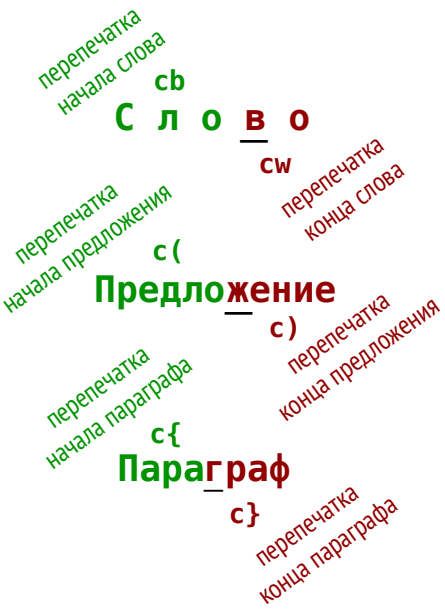
Заключительный абзац.

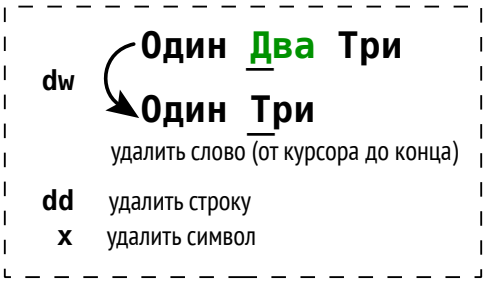
H	верх экрана
M	центр экрана
L	низ экрана

Листание текста



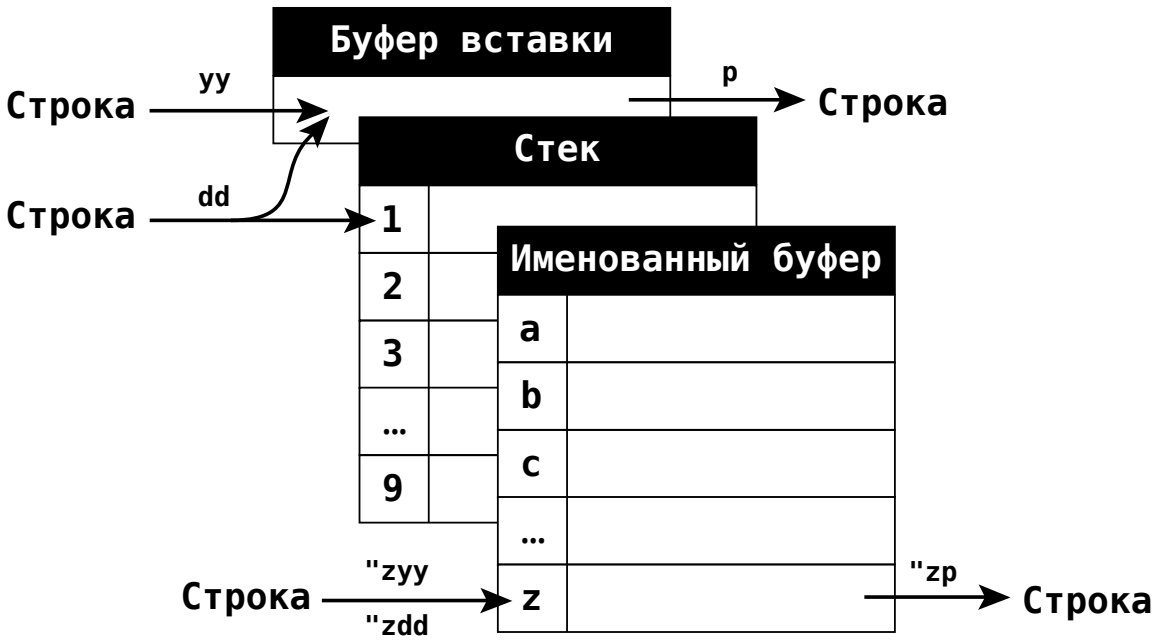
Корректировка текста





Ctrl+u	отмена последнего (одного) изменения в режиме редактирования
u	отмена последнего (одного) изменения в режиме команд
U	отмена всех изменений в ОДНОЙ строке

Буферы



Поиск / Замена текста

/abc	поиск abc к концу текста
?abc	поиск abc к началу текста
n	повторный поиск к концу текста
N	повторный поиск к началу текста

:x,y s/что/на_что/c замена текста с подтверждением

g замена текста без вопросов

от **(1)** начало

до **($)** конец

% весь текст

. текущая строка

Прочее

:set nu	нумерация строк
:set smd	показать режимы редактора
:set sm	показать парную скобку

:set ai форматирование отступами

:exu quit подсказка по команде quit

mA	установить метку A
'A	перейти к метке A
:map k s	макрос на клавишу k, выполняющий команду s. Чтобы узнать как записать k, выполнить команду cat без параметров, нажать требуемую клавишу. Первыми символами будут ^[- это клавиша Esc, следующими символами и будут те, которые надо вписать вместо k. Символ 'Enter' можно ввести, нажав Ctrl+v (появится символ ^) и далее Enter, (появится ^M)
:abb a b	автозамена при вводе текста
:!ls	выполнить команду ls и показать результат в окне редактора vi

:read !printf '\342\202\254\' вставка символа евро (в текст вставляется результат выполнения команды printf)

Настройка консоли

Драйвер консоли newcons

В FreeBSD по умолчанию используется драйвер консоли newcons (vt). Драйвер запускается и работает без каких-либо дополнительных настроек. Узнать больше о консольном драйвере можно так:

```
John@work:~% man vt
```

Драйвер позволяет изменять шрифт в консоли (консольные шрифты хранятся в /usr/share/vt/fonts). Следующая команда изменит шрифт текущей консоли на gallant:

```
John@work:~% vidcontrol -f gallant
```

Чтобы вернуть шрифт по умолчанию, можно выполнить команду:

```
John@work:~% vidcontrol -f
```

Для того, чтобы выбранный шрифт загружался при старте системы и применялся ко всем виртуальным консолям, в файл /etc/rc.conf нужно добавить строку:

```
allscreens_flags=”-f gallant”
```

Дополнительные сведения можно получить в справочной системе:

```
John@work:~% man rc.conf
```

Можно улучшить отображение символов, добавив в файл /etc/rc.conf строку (загрузка модуля для intel):

```
kld_list=”i915kms”
```

Примечание
Следует иметь в виду, что эта строка загрузит **встроенный** модуль, с которым не будут работать приложения, требующие от графики высокой производительности. Это может выражаться в следующем сообщении:

```
MESA-LOADER: failed to retrieve device information
Error! Unsupported graphics card or driver.
A graphics card and driver with support for OpenGL 3.3 or higher is required.
```

В этом случае необходимо использовать **внешний** модуль с таким же названием. Для этого устанавливаем программу **drm-kmod** (см. ниже). Для загрузки именно этого модуля, в файле /etc/rc.conf в строке kld_list=... должно быть следующее:

```
kld_list=”/boot/modules/i915kms.ko”
```

Если этого не указать, то Xorg (ведь дескоп не ограничивается лишь консолью!) будет использовать встроенный модуль, независимо от того, есть или нет упоминание i915kms в /etc/rc.conf (в выводе kldstat есть упоминание **drm2**, в случае внешнего модуля — **drm**, **linuxkpi**, **linuxkpi_gplv2**).
Поскольку в статье рассматривается построение десктопа, то приведенная выше строка должна все же присутствовать в /etc/rc.conf.

После перезагрузки драйвер установит оптимальное разрешение экрана. Если символы будут слишком мелкими, то можно загрузить шрифт по-крупнее (файл /etc/rc.conf):

```
allscreens_flags="-f terminus-b32"
```

terminus-b32 — крупный и четкий шрифт, имеющий в своем составе русские символы (в gallant, например, русских символов нет).

Выбор раскладки клавиатуры

Для переключения раскладки клавиатуры клавишами Ctrl+Shift, нужно чтобы в файле /etc/rc.conf была строка:

```
keymap="ru.win.kbd"
```

Это можно задать при установке (Keymap selection) или вручную, отредактировав /etc/rc.conf. Раскладки находятся в каталоге /usr/share/vt/keymaps.

Настройка login-класса

Описания login-классов находятся в файле /etc/login.conf. Для класса russian описание выглядит так:

```
russian|Russian Users Accounts:\
       :charset=UTF-8:\
       :lang=ru_RU.UTF-8:\
       :tc=default:
```

Редактировать его скорее всего не потребуется. Если что-то пришлось поменять, то для обновления базы выполняем команду:

```
root@work:/home/John # cap_mkdb /etc/login.conf
```

Подготовка к установке дополнительных программ

Для получения самых свежих версий программ, скорее всего, потребуется подключение к сети Интернет. Для доступа к Интернет требуется правильно настроенная сетевая карта. Если она не была настроена при установке системы, то исправить это можно при помощи команды bsdconfig:

```
root@work:/home/John # bsdconfig
```

Далее следует выбрать пункт **Networking management**. После настройки сетевой карты можно продолжить установку программ из сети. Если программы устанавливаются через компиляцию исходных текстов, то нужно иметь актуальное дерево портов. Если программы будут устанавливаться из пакетов (уже скомпилированных программ), то необходимо иметь доступ к репозиторию этих пакетов. По умолчанию, все настроено на официальные источники в сети интернет.

Как правило, но не всегда, одна и та же программа может быть установлена как из портов, так и из пакетов. Установка программы из портов может понадобится, к примеру, в том случае, если нужно собрать ее с параметрами, отличающимися от тех, с которыми был собран официальный пакет.

Получение и обновление дерева портов

Чтобы получить актуальное дерево портов, нужно запустить команду:

```
root@work:/home/John # portsnap fetch extract
```

Выполнение команды может занять некоторое время. Далее можно обновлять дерево портов до актуального состояния командой:

```
root@work:/home/John # portsnap fetch update
```

Установка пакетного менеджера

Установка программ из пакетов происходит через пакетный менеджер pkg. В свежеустановленной системе присутствует лишь одноименная утилита, которая позволяет выкачать из сети актуальную версию менеджера и установить ее.

```
root@work:/home/John # pkg
```

Добавление репозиториев

Репозитории пакетов могут быть не только официальные. Можно создать и свои, разместив их где угодно, даже на флеш-накопителе. Нужно лишь следить за актуальностью таких репозиториев. Для примера создадим описание для репозитория atHand. Создадим файл atHand.conf в каталоге **/usr/local/etc/pkg/repos/** (возможно его потребуется создать) со следующим содержимым:

```
atHand: {
        url: "file:///dist/myrepo",
        mirror_type: "none",
        enabled: yes
}
```

Теперь, при вызове пакетного менеджера, можно будет указывать имя репозитория atHand. При этом менеджер будет искать репозиторий в директории /dist/myrepo.

Удобная работа с флеш-накопителем

Монтирование флеш-накопителя вручную неудобно и требует внимания при отключении (требуется ручное размонтирование). Предлагается следующий вариант автомонтирования.

Настройка

В каталоге /usr/local/etc размещаем скрипт **usbdrive**, текст которого приведен в приложении. Если каталога /usr/local/etc нет, то создаем его:

```
root@work:/home/John # mkdir /usr/local/etc
```

Назначаем права на этот файл:

```
root@work:/home/John # chmod 500 /usr/local/etc/usbdrive
```

В каталоге /usr/local/etc/rc.d размещаем скрипт **usb_mount_clean.sh**, текст которого также приводится в приложении. Если каталога /usr/local/etc/rc.d нет, то создаем его:

```
root@work:/home/John # mkdir /usr/local/etc/rc.d
```

В файл /etc/devd.conf добавляем следующие строки:

```
attach 10 {
        match "device-name" "umass[0-9]+";
        action "sleep 4 && /usr/local/etc/usbdrive mount";
};

detach 10 {
        match "device-name" "umass[0-9]+";
        action "/usr/local/etc/usbdrive umount";
};
```

Перезапустим devd:

```
root@work:/home/John # /etc/rc.d/devd restart
```

Как пользоваться?

Скрипт **usbdrive** может смонтировать разделы FAT32 на чтение-запись для пользователей группы operator. При подключении флеш-накопителя в каталоге /media появится каталог с именем, аналогичным имени файла в каталоге /dev (da0s1, например). Также в каталоге /media появится ссылка (A, B и т. п.), указывающая на этот каталог. Теперь прочитаем содержимое:

```
John@work:~% ls /media/A/
```

Для дополнительного удобства можно создать файл /media/.usb_device.lst такого вида:

```
XXXXXXXXXXXXX    ABC
ZZZZZZZZZZZZZ    DEF
…
```

где

```
XXXXXXXXXXXXX
ZZZZZZZZZZZZZ
```

серийные номера флеш-накопителей, а

```
ABC
DEF
```

желаемые имена этих накопителей (взамен A, B и т. п.).

Примечание

Скрипт **usbdrive** монтирует накопитель в синхронном режиме, что сказывается на производительности. Для ускорения работы можно заменить строку монтирования (убрать **-o sync**):

```
# Монтирование раздела (msdosfs, sync)
/sbin/mount_msdosfs -o sync -L ru_RU.UTF-8 /dev/$part $workdir/$part()
```

Это позволит намного ускорить работу с накопителем, но при его извлечении необходимо завершать все отложенные операции записи, например, так:

```
John@work:~% sync
```

После завершения работы команды sync, накопитель можно извлечь.

Как определить серийный номер флеш-накопителя?

Подключаем флеш-накопитель и узнаем имя устройства:

```
John@work:~% dmesg -a
```

Выполняем команду (например, для da0):

```
root@work:/home/John # camcontrol inquiry da0 -S
```

Этот серийный номер можно использовать в /media/.usb_device.lst, но вручную его вписывать не потребуется, так как серийные номера подключенных флеш-накопителей будут записываться в файл /media/.usb_device.lst автоматически по мере подключения устройств. Если такой функциональности не требуется, то нужно просто удалить файл /media/.usb_device.lst.

Конструирование десктопа

Настройка графического интерфейса

Xorg

Установка

```
root@work:/home/John # pkg install Xorg
```

Можно сразу установить видеодрайвер. Например, для intel:

```
root@work:/home/John # pkg install xf86-video-intel
```

и DRM-модуль, управляющий GPU:

```
root@work:/home/John # pkg install drm-kmod
```

Без этого модуля не будут работать программы, требующие высокопроизводительной графики (blender, например). Xorg запускаться и работать будет и без этого модуля (так как при необходимости он загрузит встроенный модуль).

Настройка

```
root@work:/home/John # X -configure
root@work:/home/John # cp -v /root/xorg.conf.new /etc/X11/xorg.conf
```

В файл /etc/X11/xorg.conf вносим следующие изменения:
В секцию ServerLayout добавляем строку:

```
Option "AutoAddDevices" "false"
```

В секцию InputDevice (для Keyboard0) добавляем строки:

```
Option "XkbRules"   "xorg"
Option "XkbLayout"  "us,ru(winkeys)"
Option "XkbOptions" "grp:ctrl_shift_toggle, grp_led:scroll"
```

В секцию Files добавляем строку:

```
FontPath "/usr/local/share/fonts/personal/"
```

Для дополнительных шрифтов создадим директорию /usr/local/share/fonts/personal/:

```
root@work:/home/John # mkdir /usr/local/share/fonts/personal/
```

В эту директорию будем копировать шрифты, которые нравятся и которых нет в системе. Для того, чтобы шрифты можно было использовать, выполним команды:

```
root@work:/home/John # mkfontscale /usr/local/share/fonts/personal
```

```
root@work:/home/John # mkfontdir /usr/local/share/fonts/personal
```

Некоторые шрифты есть в пакетах. Установим шрифт **terminus**:

```
root@work:/home/John # pkg install terminus-font terminus-ttf
```

Отображение шрифтов на экране

Настроить режим отображения шрифтов можно в файле **/usr/local/etc/fonts/local.conf** (для всех пользователей) или в файле **.config/fontconfig/fonts.conf** в домашней директории (для конкретного пользователя). Пример:

```
<?xml version="1.0"?><!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
<dir>~/.fonts</dir>
<match target="font">
       <edit name="lcdfilter" mode="assign"><const>lcddefault</const></edit>
       <edit name="autohint" mode="assign"><bool>false</bool></edit>
       <edit name="hinting" mode="assign"><bool>true</bool></edit>
       <edit name="hintstyle" mode="assign"><const>hintfull</const></edit>
       <edit name="rgba" mode="assign"><const>rgb</const></edit>
</match>
<match target="font">
       <test name="size" compare="less">
              <double>16</double>
       </test>
       <edit name="antialias" mode="assign"><bool>false</bool></edit>
</match>
<match target="font">
       <test name="pixelsize" compare="less" qual="any">
              <double>30</double>
       </test>
       <edit name="antialias" mode="assign"><bool>false</bool></edit>
</match>
<match target="font">
       <test qual="any" name="family" compare="contains">
              <string>pfennig</string>
       </test>
       <edit name="antialias" mode="assign"><bool>true</bool></edit>
</match>
<match target="font">
       <test qual="any" name="family" compare="contains">
              <string>century gothic</string>
       </test>
       <edit name="antialias" mode="assign"><bool>false</bool></edit>
</match>
<match target="font">
       <test qual="any" name="family" compare="contains">
              <string>arial</string>
       </test>
       <edit name="antialias" mode="assign"><bool>false</bool></edit>
</match>
<match target="font">
       <test qual="any" name="family" compare="contains">
              <string>tahoma</string>
       </test>
       <edit name="antialias" mode="assign"><bool>false</bool></edit>
</match>
<match target="font">
       <test qual="any" name="family" compare="contains">
              <string>times new roman</string>
       </test>
```

```
        <edit name="antialias" mode="assign"><bool>false</bool></edit>
  </match>
  <match target="font">
        <test qual="any" name="family" compare="contains">
              <string>verdana</string>
        </test>
        <edit name="antialias" mode="assign"><bool>false</bool></edit>
  </match>
  <alias>
        <family>serif</family>
        <prefer>
              <family>Times New Roman</family>
              <family>PT Serif</family>
        </prefer>
  </alias>
  <alias>
        <family>sans-serif</family>
        <prefer>
              <family>Arial</family>
              <family>PT Sans</family>
        </prefer>
  </alias>
  <alias>
        <family>monospace</family>
        <prefer>
              <family>Courier New</family>
              <family>PT Mono</family>
        </prefer>
  </alias>
  </fontconfig>
```

Узнать больше о fontconfig можно так:

```
John@work:~% man fonts-conf
```

Шрифты в приложениях

Существуют Qt-приложения и gtk-приложения. Шрифты в интерфейсе Qt-приложений (libreoffice, qtcreator, ...) можно настроить при помощи программы **qt5ct**. Далее необходимо задать переменную окружения **QT_QPA_PLATFORMTHEME**. Например, в файле .cshrc для tcsh:

```
setenv  QT_QPA_PLATFORMTHEME    qt5ct
```

Шрифты в интерфейсе для gtk-приложений (gimp, ...) можно задать в файле **.gtkrc-2.0** в каталоге пользователя:

```
style "user-font"
{
        font_name="Pfennig 12"
}
widget_class "*" style "user-font"

gtk-font-name = "Pfennig 12"
```

Для gtk-приложений (geany, firefox, geeqie, inkscape, ...) можно создать файл **.config/gtk-3.0/settings.ini** в каталоге пользователя.

Например:

```
[Settings]
gtk-font-name = Pfennig 12
```

Оконный менеджер Fluxbox

Установка

Установка из портов понадобится в том случае, если возникнет необходимость украсить пункты меню пиктограммами. Для этого необходимо собрать Fluxbox с поддержкой библиотеки imlib2:

```
root@work:/home/John # cd /usr/ports/x11-wm/fluxbox && make install clean
```

В появившемся окне конфигурирования следует отметить пункт IMLIB2.
Если этого не требуется, то можно установить оконный менеджер пакетом:

```
root@work:/home/John # pkg install fluxbox
```

Дополнительно можно установить композитный менеджер xcompmgr (используется для придания окнам прозрачности), просмотрщик изображений feh (используется для помещения картинки на рабочий стол в качестве фона) и индикатор клавиатурных раскладок xxkb.

```
root@work:/home/John # pkg install xcompmgr feh xxkb
```

Настройка FluxBox

Пользовательские настройки Fluxbox хранятся в домашней директории пользователя в поддиректории .fluxbox. Изменять настройки можно прямо в них, а можно через графический интерфейс в самом Fluxbox. Можно сперва произвести настройку, а затем скопировать директорию .fluxbox в другое место. В будущем, для настройки будет достаточно скопировать сохраненную директорию .fluxbox в домашюю директорию пользователя.

Настройка xxkb

Настройки хранятся в файле .xxkbrc.
Пример:

```
XXkb.mainwindow.type: tray

XXkb.mainwindow.enable: yes
XXkb.mainwindow.geometry: 28x28+0+2
XXkb.mainwindow.label.enable: yes
XXkb.mainwindow.label.font: -xos4-terminus-medium-r-normal--16-160-72-72-c-80-
iso10646-1
XXkb.mainwindow.label.text.1: En
XXkb.mainwindow.label.text.2: Ru
XXkb.mainwindow.label.background: gray50

XXkb.button.enable: yes
XXkb.button.geometry: 28x28+34+2
XXkb.button.label.enable: yes
XXkb.button.label.font: -xos4-terminus-medium-r-normal--16-160-72-72-c-80-iso10646-1
```

```
XXkb.button.label.text.1: En
XXkb.button.label.text.2: Ru
XXkb.button.label.background: gray50

XXkb.app_list.wm_class_name.ignore: xcalc xclock
XXkb.ignore.reverse
```

Запуск

Создадим в домашней директории файл .xinitrc со следующим содержимым:

```
xset -b
fbsetroot -solid black
xcompmgr &
xxkb &
exec fluxbox
```

Теперь можно смотреть результат:

```
John@work:~% startx
```

Если все прошло успешно, то можно заменить видеодрайвер vesa на более производительный. Для этого устанавливаем видеодрайвер и DRM-модуль, если они еще не были установлены. Для intel выполним команду:

```
root@work:/home/John # pkg install xf86-video-intel drm-kmod
```

Далее в файле /etc/X11/xorg.conf в секции "Device" изменим параметр "Driver":

```
Driver      "intel"
```

Узнать больше о драйвере intel можно так:

```
John@work:~% man intel
```

Примечание
Если видеодрайвер не удается подобрать, то можно использовать драйверы без аппаратного ускорения: vesa или modesetting, которые встроены в Xorg . Для этого в файле /etc/X11/xorg.conf в секции "Device" изменим параметр "Driver". Например так:

```
Driver      "modesetting"
```

Некоторым приложениям требуется дополнительный доступ к ресурсам, связанным с видеосистемой. Чтобы пользователь смог запустить такие приложения, добавим его в группу **video**:

```
root@work:/home/John # pw groupmod video -m John
```

Графическое приглашение на вход в систему

Если запуск оконного менеджера прошел успешно, то можно установить менеджер дисплеев xdm.

Установка

```
root@work:/home/John # pkg install xdm
```

Настройка

Конфигурация xdm содержится ф файле /usr/local/etc/X11/xdm/Xresources.
Пример:

```
xlogin*login.translations: #override \
       Ctrl<Key>R: abort-display()\n\
       <Key>F1: set-session-argument(failsafe) finish-field()\n\
       <Key>Delete: delete-character()\n\
       <Key>Left: move-backward-character()\n\
       <Key>Right: move-forward-character()\n\
       <Key>Home: move-to-begining()\n\
       <Key>End: move-to-end()\n\
       Ctrl<Key>KP_Enter: set-session-argument(failsafe) finish-field()\n\
       <Key>KP_Enter: set-session-argument() finish-field()\n\
       Ctrl<Key>Return: set-session-argument(failsafe) finish-field()\n\
       <Key>Return: set-session-argument() finish-field()
xlogin*greeting: Welcome to CLIENTHOST
xlogin*fail: Login incorrect or forbidden by policy
xlogin*echoPasswd: true

xlogin*greetFace:  xos4 terminus-12
xlogin*face:       xos4 terminus-12
xlogin*promptFace: xos4 terminus-12
xlogin*failFace:   xos4 terminus-12

xlogin*borderWidth:      0
xlogin*frameWidth:       0
xlogin*innerFramesWidth: 0

xlogin*inpColor:   black
xlogin*shdColor:   gray80
xlogin*hiColor:    black
xlogin*background: black
xlogin*foreground: gray80
xlogin*greetColor: gray80
xlogin*failColor:  gray80
*Foreground:       gray80
```

Для запуска требуется отредактировать файл /etc/ttys (изменить строку, описывающую ttyv8):

```
...
ttyv8 "/usr/local/bin/xdm -nodaemon" xterm on secure
...
```

Также следует создать файл .xsession с содержимым, аналогичным .xinirc:

```
John@work:~% cp -v ~/.xinitrc ~/.xsession
```

После установки xdm потребуется перезагрузка компьютера. Если будет появляться окно xconsole, то его можно убрать, отредактировав файл /usr/local/etc/X11/xdm/Xsetup_0 (закомментировать строку 'xconsole ...').

Изменение разрешения экрана и частоты обновления

В начале нужно получить сведения о нужном режиме:

```
John@work:~% gtf 1280 1024 75
```

Полученные строки нужно вписать в файл /etc/X11/xorg.conf в секцию Monitor, а в секцию Screen в нужный Display добавить строчку:

```
Modes “1280x1024_75.00”
```

Вписать в файл результат выполнения команды gtf можно запустив ее прямо в vi:
Последовательно набираем **:read !gtf 1280 1024 75**

Если разрешение экрана «сбилось», то вернуть его обратно можно командой xrandr:

```
John@work:~% xrandr
Screen 0: minimum 320 x 200, current 1280 x 1024, maximum 8192 x 8192
VGA1 connected 1280x1024+0+0 (normal left inverted right x axis y axis) 337mm x 270mm
   1280x1024     60.02*+  75.02
   1152x864      75.00
   1024x768      75.08    70.07    60.00
   832x624       74.55
   800x600       72.19    75.00    60.32
   640x480       75.00    72.81    60.00
   720x400       70.08
HDMI1 disconnected (normal left inverted right x axis y axis)
DP1 disconnected (normal left inverted right x axis y axis)

John@work:~% xrandr --output VGA1 --auto
```

Внешний вид программы Xterm

Можно улучшить внешний вид программы Xterm. Для этого в файле .Xdefaults (в домашней директории) добавить такие строки:

```
XTerm*scrollBar:   False
XTerm*saveLines:   1000
XTerm*background:  grey15
XTerm*foreground:  grey75
XTerm*cursorColor: lightgreen
XTerm*geometry:    80x28
XTerm*faceName:    xos4 terminus
XTerm*faceSize:    12
```

Установка приложений

Для примера установим файловый менеджер Midnight Commander (mc):

```
root@work:/home/John # pkg install mc
```

Другие десктопные приложения:

Geany	geany	текстовый редактор с подсветкой синтаксиса
LibreOffice	libreoffice	офисный пакет
GIMP	gimp	растровый графический редактор
Inkscape	inkscape	векторный графический редактор
Geeqie	geeqie	программа для просмотра изображений
FBReader	fbreader	программа для чтения электронных книг в различных форматах
Zathura	zathura zathura-djvu zathura-pdf-poppler zathura-ps	программа для просмотра файлов в форматах PDF, DjVu, PostScript
Yacas	yacas	система компьютерной алгебры
SpeedCrunch	speedcrunch	калькулятор
Mozilla Firefox	firefox	веб-браузер
Mozilla Thunderbird	thunderbird	программа для работы с электронной почтой

Загрузочный флеш-накопитель

Удобнее всего для этих целей иметь отдельный накопитель небольшого размера (например, 1 Гбайт). Для создания загрузочного флеш-накопителя потребуется файл образа, например, FreeBSD-12.1-RELEASE-amd64-memstick.img, взятый с официального сайта. Пусть он будет размещен в домашнем каталоге пользователя. Следующей командой можно развернуть этот образ на флеш-накопителе с именем da0:

```
root@work:/home/John # dd if=FreeBSD-12.1-RELEASE-amd64-memstick.img of=/dev/da0 bs=1m
conv=sync
```

Флеш-накопитель готов. С него можно загружаться и устанавливать систему.

Персональный репозиторий на флеш-накопителе

Для этих целей лучше иметь накопитель большого размера. Для создания репозитория понадобится подключение к сети.

Подготовка

Подготовим флеш-накопитель (например, da0):
Удаляем имеющиеся разделы и разметку:

```
root@work:/home/John # gpart delete -i 1 da0
root@work:/home/John # gpart destroy da0
```

Создаем разметку и разделы:

```
root@work:/home/John # gpart create -s GPT da0
root@work:/home/John # gpart add -t freebsd-ufs -l atHand da0
root@work:/home/John # newfs /dev/da0p1
```

Монтировать накопитель можно будет командой:

```
root@work:/home/John # mount /dev/gpt/atHand /dist
```

Создадим временный каталог:

```
John@work:~ % mkdir myrepo
```

Создадим файл со списком нужных программ:

```
John@work:~ % vi apps
```

Пример:

```
#
# Список приложений
#

       pkg

# Графическое окружение

       xorg
       xf86-video-intel
       drm-kmod
       xdm
       xcompmgr
       fluxbox
       xxkb
       terminus-font
       terminus-ttf
       feh

# Десктопные приложения

       geany
       libreoffice
       ru-libreoffice
       inkscape
       gimp
       krita
       blender
       geeqie
       zathura
       zathura-djvu
       zathura-pdf-mupdf
       zathura-pdf-poppler
       zathura-ps
       fbreader
       yacas
       speedcrunch
       firefox
       thunderbird

# Системные утилиты

       mc
       portmaster
       sudo
       smartmontools
       p7zip
       fusefs-ntfs
       scalpel
       testdisk

# Программирование

       qt5
       qt5ct
       qtcreator

       cmake
       cmake-gui
       ninja
```

```
# Работа с изображениями

       dcraw
       p5-Image-ExifTool
       ImageMagick

# Печать/сканирование

       cups
       gutenprint
       hplip

       sane-frontends

# Проигрыватели

       mplayer
       mpv

# Игры

       aisleriot
       simsu

       wine
       i386-wine
       wine-gecko
       wine-mono

# Разное

       stellarium
```

В каталоге пользователя создадим скрипт miniRepo (текст скрипта в приложении).

Загрузка пакетов из официального репозитория

Для загрузки достаточно запустить скрипт:

```
root@work:/home/John # ./miniRepo apps myrepo
```

В каталоге /home/John/myrepo/ будет создан готовый для использования репозиторий. Из него можно будет установить любое приложение, упоминаемое в файле apps. Дополнительные пакеты, необходимые для работы этих приложений (т.е. зависимости), также будут присутствовать в созданном репозитории.

Поместим файлы репозитория в удобное место (например, на флеш-накопитель, смонтированный в каталог /dist):

```
John@work:~ % cp -Rv myrepo /dist/
```

Использовать репозиторий можно через описание atHand (см. ранее по тексту).

Использование персонального репозитория

Подключить флеш-накопитель к компьютеру и смонтировать его в каталог /dist

```
root@work:/home/John # mount /dev/gpt/atHand /dist
```

Установить программу (например, mc):

```
root@work:/home/John # pkg install -r atHand mc
```

Размонтировать накопитель:

```
root@work:/home/John # umount /dist
```

Запустить установленное приложение:

```
John@work:~ % mc
```

Синхронизация персонального репозитория с официальным

Синхронизация репозитория, находящегося в директории myrepo, с официальным репозиторием осуществляется командой:

```
root@work:/home/John # ./miniRepo apps myrepo
```

При этом будут сравнены контрольные суммы имеющихся пакетов (в директории /home/John/myrepo/All/) с суммами пакетов, находящихся в официальном репозитории. При несовпадении, имеющиеся пакеты будут заменены. Также будут добавлены новые пакеты, появившиеся в apps позднее и удалены те, которые более не упоминаются в apps. После синхронизации пакетов, репозиторий будет перестроен.

Преимущество персонального репозитория

Преимущество всего одно – возможность установить необходимые приложения на компьютер, не имеющий выход в Интернет.

Создание пакета для приложения, собранного из порта

Порты позволяют собрать программу с опциями, отличными от тех, с которыми были собраны версии из официального репозитория. Но пересобирать программы каждый раз при смене компьютера долго. Поэтому достаточно собрать нужную программу один раз, а затем, создать свой пакет. Далее этот пакет можно будет разместить в персональном репозитории, заменив им официальную версию.

Пример создания пакета для fluxbox

Сборка и установка:

```
root@work:/home/John # cd /usr/ports/x11-wm/fluxbox && make install clean
```

Создание пакета в каталоге /home/John/mypack:

```
John@work:~ % pkg create -o mypack fluxbox
```

Затем пакет (файл с расширением .txz) можно скопировать в репозиторий (в кататог All) и далее пересоздать репозиторий:

```
John@work:~ % cp -v mypack/fluxbox-1.3.7.txz myrepo/All/
John@work:~ % pkg repo myrepo
```

Как собрать дескоп за 20 минут

Чтобы настроить компьютер для работы в качестве десктопа понадобится проделать много действий вручную. Имея флеш-накопитель, подобный описанному выше, а, также, написав скрипт для автоматизации, можно производить сборку десктопа примерно за 20 минут. Текст подобного скрипта (**mpi**) приведен в приложении. Этот скрипт использует и другие скрипты (**fbMenuModify**, **usbdrive**, **usb_mount_clean.sh**), текст которых также приведен в приложении. Разместим эти скрипты, а, также вспомогательные файлы на флеш-накопителе в каталоге personal. Структура этого каталога следующая:

```
personal
  conf
    fluxbox
    xdm
  files
    fonts
       droid
       terminus
       paratype
       windows
       ...
  scripts
```

Именно такую структуру использует скрипт mpi.

Шаги для сборки десктопа

Подключить подготовленный флеш-накопитель и установить систему
После перезагрузки зарегистрироваться пользователем, созданным в процессе установки (например, John)
Выполнить команды:

```
John@work:~ % su
root@work:/home/John # mkdir /dist
root@work:/home/John # mount /dev/gpt/atHand /dist
root@work:/home/John # /dist/personal/scripts/mpi
```

После окончания работы перерегистрироваться (перезайти) пользователем (в нашем примере John).
Проверить работу Xorg командой:

```
John@work:~ % startx
```

Если запуск успешен, то включить способ входа в систему через xdm:

```
root@work:/home/John # vi /etc/ttys
```

Поправим строку:

```
...
ttyv8 "/usr/local/bin/xdm -nodaemon" xterm on secure
...
```

Далее:

```
root@work:/home/John # cp ~/.xinitrc ~/.xsession
```

Уберем окно xconsole при запуске xdm:

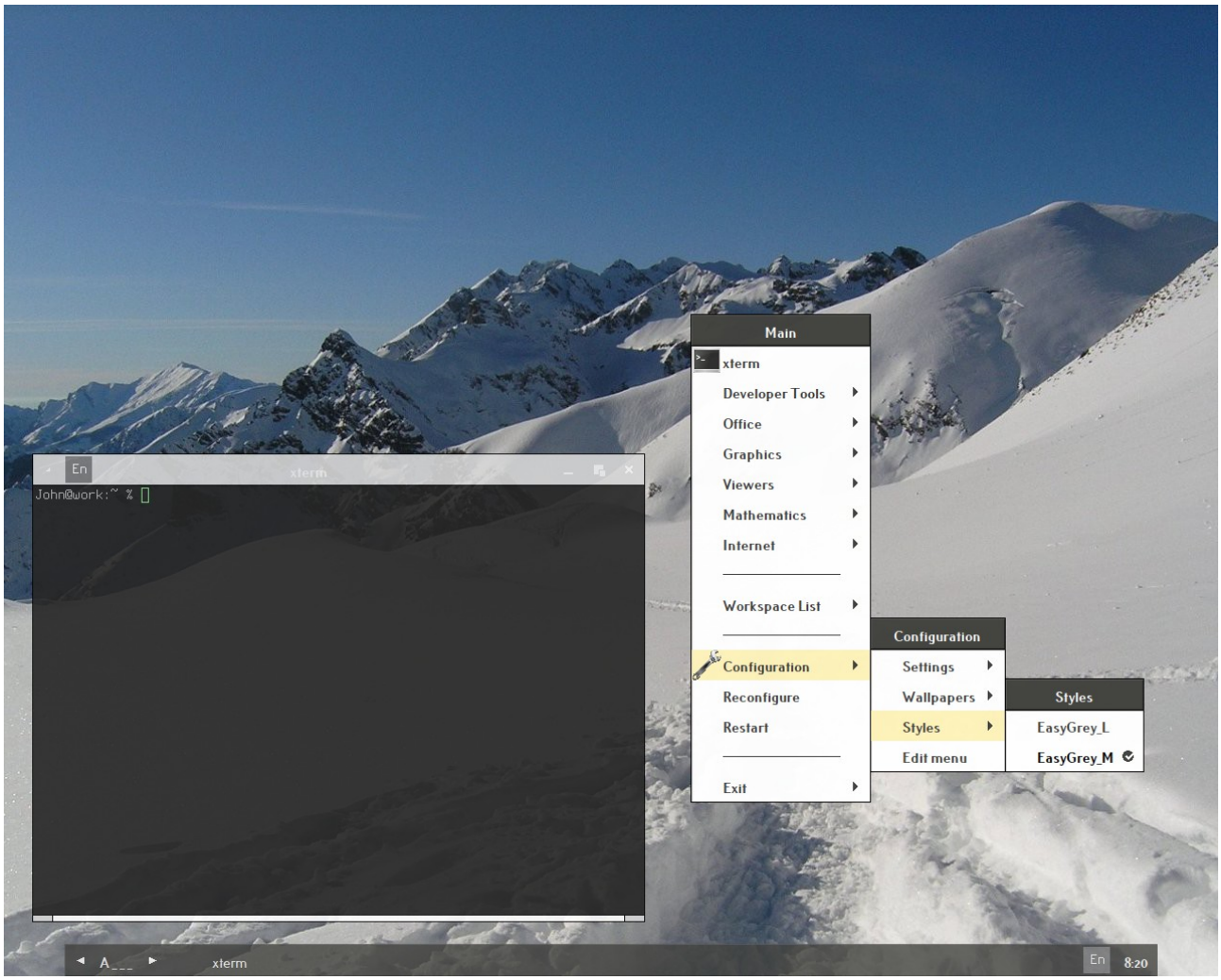
```
root@work:/home/John # vi /usr/local/etc/X11/xdm/Xsetup_0
```

закомментировав строку, начинающуюся с xconsole.

Перезагружаем компьютер:

```
root@work:/home/John # shutdown -r now
```

После регистрации в системе, получаем готовый десктоп:



Как можно разнообразить десктоп

Часы со стрелками

Добавим на рабочий стол часы со стрелками. Будем использовать программу **xclock**. Настроим автоматический запуск. Для этого в файл ~/.xsession, до запуска fluxbox, добавим строку:

```
xclock &
```

Настроим оформление окна для xclock. Для этого в файл ~/.fluxbox/apps добавим строки:

```
[app] (name=xclock)
  [Position]    (TOPRIGHT)      {10 10}
  [Dimensions]  (50 50}
  [Deco]        {BORDER}
  [Hidden]      {yes}
  [Sticky]      {yes}
[end]
```

Календарь

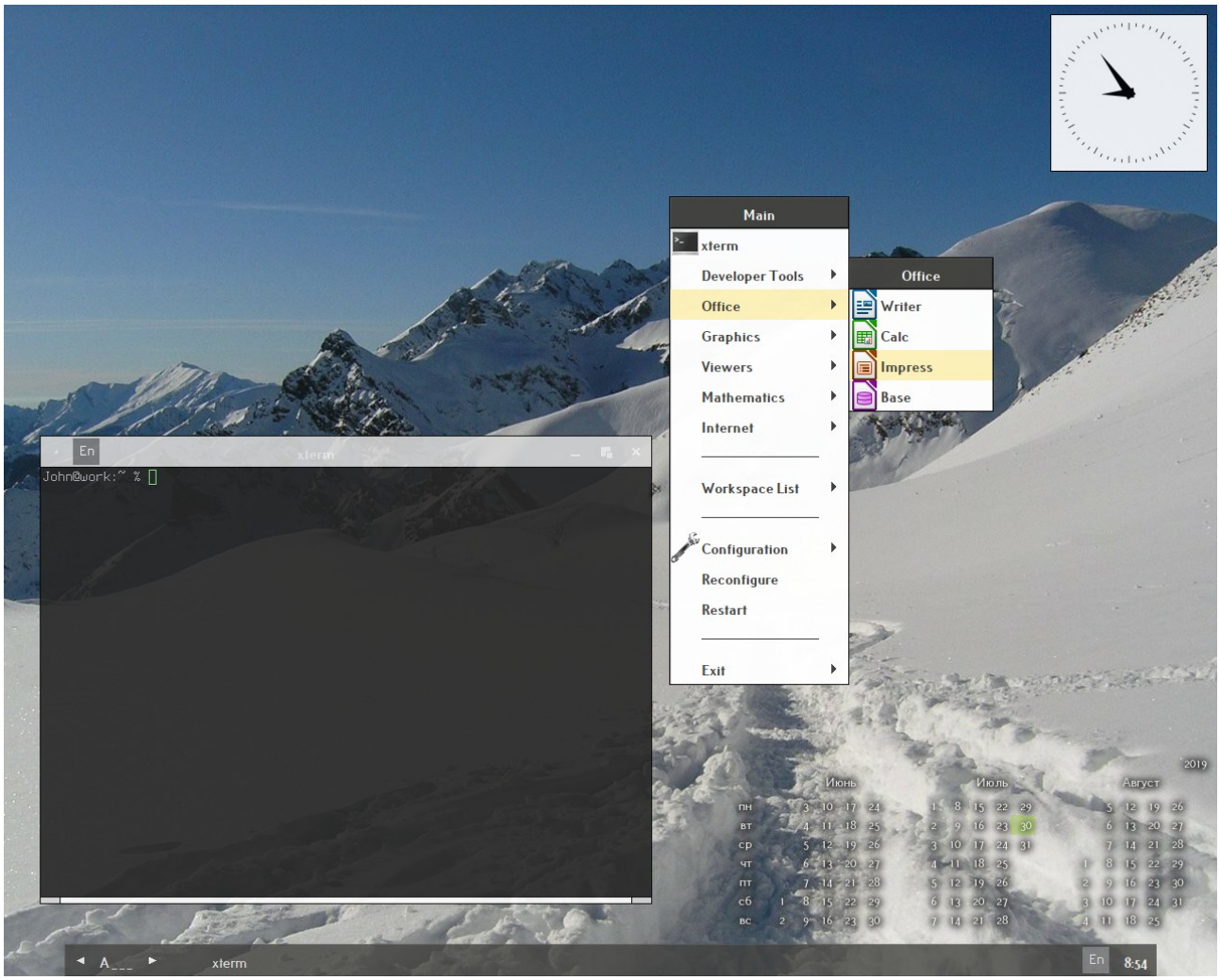
Фоном рабочего стола выбрано изображение track.jpg. Нанесем на него календарь. Для формирования изображения с календарем, применим скрипт **wpwc**. Его текст приведен в приложении. Для его работы нужны утилиты из пакета ImageMagick. Разместим wpwc в каталоге ~/mysh/. Изображение будем формировать перед запуском fluxbox. Внесем следующую строку в файл ~/.xsession:

```
~/mysh/wpwc ~/.fluxbox/wallpapers/track.jpg ~/.fluxbox/wallpapers/wpwc.jpg
```

Затем, через главное меню FluxBox установим фон рабочего стола:

```
Configuration → Wallpapers → wpwc.jpg
```

В итоге:



Приложение

Скрипт mpi

```
#!/bin/sh
#
# Script name    : FreeBSD personal set (offline)
# Version        : 3.0 / for FreeBSD 12.1
# Date           : 14.08.2020
#

personalDir='/dist/personal'
tempFile=~/file.temporary

cn='\033[33m'
cf='\033[m'

setValue(){
        sed "/$2/d" $1 > $tempFile
        echo $2=$3 >> $tempFile
        mv $tempFile $1
}

printf "\n${cn}FreeBSD personal setup${cf}\n\n"

# [A] Настройка консоли

# [A1] Новый драйвер консоли 'newcons'

printf "${cn}Setting parameters for 'newcons'...${cf}\n"

fileConfig='/boot/loader.conf'

#setValue $fileConfig hw.vga.textmode 0
#setValue $fileConfig kern.vty vt

fileConfig='/etc/rc.conf'

module='/boot/modules/i915kms.ko'
grep -q "${module}" $fileConfig
if [ $? -ne 0 ]
then
        grep -q "^kld_list=" $fileConfig
        if [ $? -ne 0 ]
        then
                echo "kld_list=\"$module\"" >> $fileConfig
        else
                sed "/^kld_list="\"".*"\""/{
                s%"\""% $module"\""%2
                s/[[:blank:]]\{1,\}/ /g;
                s/"\"" /"\""/1;
                }" $fileConfig > $tempFile
                mv $tempFile $fileConfig

                sed "/^kld_list="\"".*\\\[[:blank:]]*$/a\\
          $module\\\\
                " $fileConfig > $tempFile
                mv $tempFile $fileConfig
        fi
fi

fileConfig=

# [A2] Выбор раскладки клавиатуры

printf "${cn}Set russian keymap...${cf}\n"

fileConfig='/etc/rc.conf'

grep -q "keymap=" $fileConfig
if [ $? -ne 0 ]
then
```

```
        setValue $fileConfig keymap \"ru.win.kbd\"
fi

fileConfig=

# [A3] Настройка login-класса

loginClassName=russian
loginClassDescription='Russian Users Accounts'
charset=UTF-8
lang=ru_RU.UTF-8

fileConfig='/etc/login.conf'

grep -q "^$loginClassName|" $fileConfig
if [ $? -ne 0 ]
then
        printf "${cn}Create ${loginClassName} login class...${cf}\n"
        echo "${loginClassName}|${loginClassDescription}:\\" >> $fileConfig
        echo "        :charset=${charset}:\\" >> $fileConfig
        echo "        :lang=${lang}:\\" >> $fileConfig
        echo "        :tc=default:" >> $fileConfig
else
        printf "${cn}Login class '${loginClassName}' already exists...${cf}\n"
        # printf "${cn}Modify ${loginClassName} login class...${cf}\n"
        # sed "/^${loginClassName}|/,/tc=default/{
        # s/\(charset=\).*\(:\)/\1${charset}\2/;
        # s/\(lang=\).*\(:\)/\1${lang}\2/
        # }" $fileConfig > $tempFile
        # mv $tempFile $fileConfig
fi

cap_mkdb $fileConfig

fileConfig=

# [A4] Настройка автомонтирования USB-накопителей

if [ ! -d /usr/local/etc ]
then
        mkdir /usr/local/etc
fi
cp ${personalDir}/scripts/usbdrive /usr/local/etc/
chmod 500 /usr/local/etc/usbdrive

if [ ! -d /usr/local/etc/rc.d ]
then
        mkdir /usr/local/etc/rc.d
fi
cp ${personalDir}/scripts/usb_mount_clean.sh /usr/local/etc/rc.d

attachAction='/usr/local/etc/usbdrive mount'
detachAction='/usr/local/etc/usbdrive umount'

fileConfig='/etc/devd.conf'

sed -n "/^[[:space:]]*attach/,/};/{
/^[[:blank:]]*match[[:blank:]]*\"device-name\"[[:blank:]]*\"umass\[0-9\]+\";/,/};/p
}" $fileConfig | grep -q "^[[:blank:]]*action[[:blank:]]"

if [ $? -ne 0 ]
then
        printf "${cn}Create action for attach usb drive...${cf}\n"
        echo >> $fileConfig
        echo 'attach 10 {' >> $fileConfig
        echo '        match "device-name" "umass[0-9]+";' >> $fileConfig
        echo "        action \"sleep 4 && ${attachAction}\";" >> $fileConfig
        echo '};' >> $fileConfig
else
        printf "${cn}Modify action for attach usb drive...${cf}\n"
        sed "/^[[:space:]]*attach/,/};/{
        /^[[:blank:]]*match[[:blank:]]*\"device-name\"[[:blank:]]*\"umass\[0-9\]+\";/,/};/{
        /^[[:blank:]]*action/{
        a\\
        action \"sleep 4 && $attachAction\";
        d
        }
        }
        }" $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi
```

```
sed -n "/^[[:space:]]*detach/,/};/{
/^[[:blank:]]*match[[:blank:]]*\"device-name\"[[:blank:]]*\"umass\[0-9\]+\";/,/};/p
}" $fileConfig | grep -q "^[[:blank:]]*action[[:blank:]]"

if [ $? -ne 0 ]
then
        printf "${cn}Create action for detach usb drive...${cf}\n"
        echo >> $fileConfig
        echo 'detach 10 {' >> $fileConfig
        echo '        match "device-name" "umass[0-9]+";' >> $fileConfig
        echo "        action \"${detachAction}\";" >> $fileConfig
        echo '};' >> $fileConfig
else
        printf "${cn}Modify action for detach usb drive...${cf}\n"
        sed "/^[[:space:]]*detach/,/};/{
        /^[[:blank:]]*match[[:blank:]]*\"device-name\"[[:blank:]]*\"umass\[0-9\]+\";/,/};/{
        /^[[:blank:]]*action/{
        a\\
        action \"$detachAction\";
        d
        }
        }
        }" $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi

fileConfig=

# [A5] Установка пакетного менеджера

printf "${cn}Install utility 'pkg'...${cf}\n"
env SIGNATURE_TYPE=none pkg add /dist/myrepo/All/pkg-[0-9]*.txz

# [A6] Конфигурации репозиториев

printf "${cn}Preparing to use packages...${cf}\n"

mkdir -p /usr/local/etc/pkg/repos

fileConfig='/usr/local/etc/pkg/repos/atHand.conf'

if [ -f $fileConfig ]
then
        printf "${cn}...configuration for personal repository is already exists${cf}\n"
else
        printf "${cn}...create configuration for personal repository${cf}\n"
        cp ${personalDir}/conf/atHand.conf $fileConfig
fi

fileConfig=

# [B] Настройка графического окружения

# [B1] Установка и настройка Xorg

printf "${cn}Install Xorg...${cf}\n"
pkg install -y -r atHand Xorg xdm xcompmgr
pkg install -y -r atHand xf86-video-intel drm-kmod

fileConfig='/etc/X11/xorg.conf'

if [ -f $fileConfig ]
then
        printf "${cn}...configuration for Xorg is already exists${cf}\n"
else
        printf "${cn}...create configuration for Xorg${cf}\n"
        /usr/local/bin/X -configure
        cp /root/xorg.conf.new $fileConfig
        sed '/Section "ServerLayout"/,/EndSection/{/EndSection/i\
        Option         "AutoAddDevices" "false"
        }' $fileConfig > $tempFile
        mv $tempFile $fileConfig
        sed '/Section "InputDevice"/,/EndSection/{/"kbd"/a\
        Option      "XkbRules"   "xorg"\
        Option      "XkbLayout"  "us,ru(winkeys)"\
        Option      "XkbOptions" "grp:ctrl_shift_toggle, grp_led:scroll"
        }' $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi

printf "${cn}...add some fonts for Xorg${cf}\n"
```

```
pkg install -y -r atHand terminus-font terminus-ttf

sed -n '/Section "Files"/,/EndSection/p' $fileConfig | grep -q
'^[[:blank:]]*FontPath[[:blank:]]*"/usr/local/share/fonts/terminus-font/"'

if [ $? -ne 0 ]
then
        sed '/Section "Files"/,/EndSection/{/EndSection/i\
        FontPath     "/usr/local/share/fonts/terminus-font/"
        }' $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi

sed -n '/Section "Files"/,/EndSection/p' $fileConfig | grep -q
'^[[:blank:]]*FontPath[[:blank:]]*"/usr/local/share/fonts/TerminusTTF/"'

if [ $? -ne 0 ]
then
        sed '/Section "Files"/,/EndSection/{/EndSection/i\
        FontPath     "/usr/local/share/fonts/TerminusTTF/"
        }' $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi

sed -n '/Section "Files"/,/EndSection/p' $fileConfig | grep -q
'^[[:blank:]]*FontPath[[:blank:]]*"/usr/local/share/fonts/personal/"'

if [ $? -ne 0 ]
then
        sed '/Section "Files"/,/EndSection/{/EndSection/i\
        FontPath     "/usr/local/share/fonts/personal/"
        }' $fileConfig > $tempFile
        mv $tempFile $fileConfig
fi

fontDir='/usr/local/share/fonts/personal'
if [ ! -d $fontDir ]
then
        mkdir $fontDir
fi
cp ${personalDir}/files/fonts/*/* $fontDir

mkfontscale $fontDir
mkfontdir $fontDir

fileConfig=

fileConfig='/usr/local/etc/fonts/local.conf'
if [ -f $fileConfig ]
then
        printf "${cn}...FontConfig is already configured${cf}\n"
else
        printf "${cn}...create configuration for FontConfig${cf}\n"
        cp ${personalDir}/conf/fonts.conf $fileConfig
fi
fileConfig=

fileConfig='/usr/local/etc/X11/xdm/Xresources'
printf "${cn}...copy personal configuration for Xdm${cf}\n"
cp ${personalDir}/conf/xdm/Xresources $fileConfig
fileConfig=

fileConfig="/home/$USER/.Xdefaults"
if [ -f $fileConfig ]
then
        printf "${cn}...configuration for X-applications is already exists${cf}\n"
else
        printf "${cn}...create configuration for X-applications${cf}\n"
        cp ${personalDir}/conf/.Xdefaults $fileConfig
fi
fileConfig=

# [B2] Установка и настройка FluxBox

printf "${cn}Install FluxBox...${cf}\n"
pkg install -y -r atHand feh fluxbox xxkb

mkdir /home/$USER/.fluxbox
cp -R ${personalDir}/conf/fluxbox/* /home/$USER/.fluxbox
cp /home/$USER/.fluxbox/mymenu.empty /home/$USER/.fluxbox/menu

fileConfig="/home/$USER/.xinitrc"
```

```
if [ -f $fileConfig ]
then
        printf "${cn}...user script '$fileConfig' is already exists${cf}\n"
else
        printf "${cn}...create user script '$fileConfig' for start FluxBox${cf}\n"
        echo 'xset -b' > $fileConfig
        echo 'fbsetroot -solid black' >> $fileConfig
        echo 'xcompmgr &' >> $fileConfig
        echo 'xxkb &' >> $fileConfig
        echo 'exec fluxbox' >> $fileConfig
fi
cp /home/$USER/.xinitrc /home/$USER/.xsession
fileConfig=

fileConfig="/home/$USER/.xxkbrc"
if [ -f $fileConfig ]
then
        printf "${cn}...configuration for XXkb is already exists${cf}\n"
else
        printf "${cn}...create configuration for XXkb${cf}\n"
        cp ${personalDir}/conf/.xxkbrc $fileConfig
fi
fileConfig=

pw groupmod operator -m $USER
pw groupmod video -m $USER

# [B3] Установка приложений

pkg install -y -r atHand mc

pkg install -y -r atHand geany
${personalDir}/scripts/fbMenuModify add 'Developer Tools' 'Geany' geany '~/.fluxbox/icons/32x32/geany.png'

pkg install -y -r atHand libreoffice
${personalDir}/scripts/fbMenuModify add 'Office' 'Writer' lowriter '~/.fluxbox/icons/32x32/lo-writer.png'
${personalDir}/scripts/fbMenuModify add 'Office' 'Calc' localc '~/.fluxbox/icons/32x32/lo-calc.png'
${personalDir}/scripts/fbMenuModify add 'Office' 'Impress' loimpress '~/.fluxbox/icons/32x32/lo-impress.png'
${personalDir}/scripts/fbMenuModify add 'Office' 'Base' lobase '~/.fluxbox/icons/32x32/lo-base.png'

pkg install -y -r atHand gimp inkscape

${personalDir}/scripts/fbMenuModify add 'Graphics' 'Inkscape' inkscape '~/.fluxbox/icons/32x32/inkscape.png'
${personalDir}/scripts/fbMenuModify add 'Graphics' 'The Gimp' gimp '~/.fluxbox/icons/32x32/gimp.png'

pkg install -y -r atHand geeqie fbreader
pkg install -y -r atHand zathura zathura-djvu zathura-pdf-poppler zathura-ps

${personalDir}/scripts/fbMenuModify add 'Viewers' 'Geeqie' geeqie '~/.fluxbox/icons/32x32/geeqie.png'
${personalDir}/scripts/fbMenuModify add 'Viewers' 'E-Book' FBReader
${personalDir}/scripts/fbMenuModify add 'Viewers' 'PDF' zathura

pkg install -y -r atHand yacas
${personalDir}/scripts/fbMenuModify add 'Mathematics' 'Yacas' 'xterm -e yacas'
'~/.fluxbox/icons/32x32/formula.png'
${personalDir}/scripts/fbMenuModify add 'Mathematics' 'Calculator' 'xcalc' '~/.fluxbox/icons/32x32/calc.png'

pkg install -y -r atHand thunderbird luakit firefox
${personalDir}/scripts/fbMenuModify add 'Internet' 'Browser' firefox '~/.fluxbox/icons/32x32/firefox.png'
${personalDir}/scripts/fbMenuModify add 'Internet' 'E-Mail' thunderbird '~/.fluxbox/icons/32x32/mail.png'

chown -R $USER:$USER /home/$USER/.fluxbox
chown $USER:$USER /home/$USER/.Xdefaults
chown $USER:$USER /home/$USER/.xsession
chown $USER:$USER /home/$USER/.xxkbrc
```

Скрипт fbMenuModify

```
#!/bin/sh

# Name: fbMenuModify
# Date: 12.02.2015
# Desc: Добавление или удаление пунктов меню в FluxBox

menuFile=/home/$USER/.fluxbox/menu
tempFile=~/.file.temporary

case $1 in
```

```
add)
        if [ $# -lt 4 ]
        then
                echo 'To  few parameters...'
        else
                # Добавление программы в главное меню или подменю
                # Если нужно добавить программу в главное меню, то название подменю указывается 'MAIN'
                # Если название подменю уже существует, то программа добавится в конец этого подменю
                # Если название подменю не существует, то оно будет создано и в него будет добавлена указанная
программа
                # Подменю создается в главном меню, вложенные меню можно создать только вручную
                # title - название подменю (как оно будет написано на экране)
                # name  - название программы (как оно будет написано на экране)
                # cmd   - строка запуска программы
                # icon  - графический значок для программы

                title=$2
                name=$3
                cmd=$4
                icon=$5

                if [ "$icon" ]
                then
                        menuLine="[exec] ($name) {$cmd} <$icon>"
                else
                        menuLine="[exec] ($name) {$cmd}"
                fi

                # Метки в файле menuFile, по которым идет ориентирование (они должны существовать)
                # placeItem - перед этой меткой будут добавляться программы в главное меню
                # placeSub  - перед этой меткой будут добавляться новые подменю

                placeItem='#PlaceNewItem'
                placeSub='#PlaceNewSub'

                # Команда sed: x;p;x - добавляет пустую строку перед регулярным выражением (placeItem или
placeSub)

                if [ "$title" = MAIN ]
                then
                        sed "/^$placeItem/{
                        i\\
    $menuLine
                        x;p;x
                        }" $menuFile > $tempFile
                else
                        grep -q -i "^[[:blank:]]*\[submenu\][[:blank:]]*($title)" $menuFile
                        if [ $? -ne 0 ]
                        then
                                sed "/^$placeSub/{
                                i\\
    [submenu] ($title)\\
      $menuLine\\
    [end]
                                x;p;x
                                }" $menuFile > $tempFile
                        else
                                sed "/^[[:blank:]]*\[submenu\][[:blank:]]*($title)/,/\[end\]/{/\[end\]/i\\
      $menuLine
                                }" $menuFile > $tempFile
                        fi
                fi
                mv $tempFile $menuFile
        fi
;;
del)
        if [ $# -lt 2 ]
        then
                echo 'To few parameters...'
        else
                # Удаление программы из главного меню или подменю.
                # Если в подменю не остается программ, то удаляется и подменю
                #
                # subName - название подменю, в котором находится удаляемая программа
                # Вычисляется следующим образом:
                #       Обрабатываем строки файла в обратной последовательности (от последней строчки к первой)
                #       Выбираем блок текста от строки, содержащей имя удаляемой программы до заголовка подменю
[submenu] ...
                #       Обрабатываем в этом блоке только последнюю строку
                #       В этой строке берем только то, что находится в круглых скобках - это и есть subName
                #
                # execCount - число программ в подменю subName
```

```
                # Если execCount = 1, то будет удаляться все подменю, иначе только одна строчка с программой

                name=$2
                subName=`\
                        sed '1!G;h;$!d' $menuFile | \
                        sed -n "/^[[:blank:]]*\[exec\][[:blank:]]*($name)/,/^[[:blank:]]*\[submenu\]/p" | \
                        tail -1 | \
                        sed "s/^[[:blank:]]*\[submenu\][[:blank:]]*(\(.*\))/\1/"\
                `
                execCount=`\
                        sed -n "/^[[:blank:]]*\[submenu\][[:blank:]]*($subName)/,/\[end\]/p" $menuFile | \
                        grep -c "^[[:blank:]]*\[exec\]"
                `
                if [ $execCount -eq 1 ]
                then
                        sed "/^[[:blank:]]*\[submenu\][[:blank:]]*($subName)/,/\[end\]/d" $menuFile > $tempFile
                else
                        sed "/^[[:blank:]]*\[exec\][[:blank:]]*($name)/d" $menuFile > $tempFile
                fi
                mv $tempFile $menuFile
        fi
;;
*)
echo
echo 'Usage: fbMenuModify add <menuTitle> <appName> <cmd> [<icon>]'
echo '         or'
echo '       fbMenuModify del <appName>'
echo
;;
esac

# Эта команда "превратит" несколько последовательноидущих пустых строк в одну (для порядка)
sed "/[[:alnum:]]./,/^[[:blank:]]*$/!d" $menuFile > $tempFile
mv $tempFile $menuFile
```

Скрипт miniRepo

```
#!/bin/sh

# Name: miniRepo
# Date: 14.03.2019
# Vers: 1.0
# Desc: Постороение мини-репозитория
#
# Параметры: $1 - текстовый файл со списком желаемых приложений
#                 (Приложения заносятся построчно в виде 'название' (name)
#                 или 'категория/название' (origin). Строки, начинающиеся
#                 с символа '#' игнорируются. Пустые строки также
#                 игнорируются)
#            $2 - путь к мини-репозиторию
#                 (Мини-репозиторий - локальный репозиторий, который является
#                 частью эталонного репозитория. Мини-репозиторий содержит
#                 все необходимые пакеты для установки желаемых приложений.
#                 На момент запуска скрипта может как отсутствовать -
#                 загрузятся все пакеты, так и существовать - загрузятся
#                 только отсутствующие пакеты, а ненужные удалятся)
#
# Принятые обозначения:
#            refRepo - эталонный репозиторий
#            minRepo - мини-репозиторий

# Название эталонного репозитория
repo='FreeBSD'
#repo='atHand'

manpage(){
  echo 'Usage:'
  echo '       miniRepo <file> <directory>'
  echo
  echo '       <file>      - file containing a list of preferred applications'
  echo '       <directory> - directory in which the local repository will be'
  echo '                     created/updated '
}

printf "\n%s\n\n" 'This is miniRepo'

# [1]
```

```
# Проверка входных параметров

if [ $# -ne 2 ] ; then
  manpage
  echo
  exit
fi

if [ ! -f $1 ] ; then
  echo 'Error:'
  echo "       File '$1' not found"
  echo
  exit
fi

if [ ! -d $2/All ] ; then
  echo 'Warning:'
  echo -n "       The '$2/All' directory does not exist. Create? [y/N]:"
  read ch
  case $ch in
  y)
    echo
    echo -n "Creating a directory '$2/All'..."
    mkdir -p $2/All 1>/dev/null 2>&1
    if [ $? -eq 0 ] ; then
      echo successfully
    else
      echo not successfully
      exit
    fi
    ;;
  *)
    printf "\n%s\n\n" "Execution stopped."
    exit
    ;;
  esac
fi

touch $2/All/some.tmp 1>/dev/null 2>&1
if [ $? -eq 0 ] ; then
  rm $2/All/some.tmp
else
  echo 'Error:'
  echo "      It is not possible to write anything to the '$2/All' directory"
  echo
  exit
fi


# [2]
# Сбор начальных сведений

# Получение информации о всех пакетах в refRepo в виде id|origin|name

pkg update -r $repo

if [ $? -ne 0 ] ; then
  echo
  echo 'Error:'
  echo '      Could not connect to remote repository. Execution aborted.'
  echo
  exit
fi

fPkgInRefRepoAll=allPkgInRefRepo

printf "%s\n%s" ".open /var/db/pkg/repo-${repo}.sqlite" 'select id,origin,name from packages;' | pkg shell >
$fPkgInRefRepoAll

# Получение информации об уже имеющихся пакетах в каталоге $2/All

fPkgInMinRepo=pkgInMinRepo
> $fPkgInMinRepo

echo
echo "Inventory of existing packages in '$2/All' (it will take some time):"
path=`pwd`

c=`ls $2/All/ | grep "\.txz$" | wc -l`
if [ $c -ne 0 ] ; then
  cd $2/All
  sha256 -r *.txz | sort -o $path/$fPkgInMinRepo
```

```
  cd $path
  echo "There are $c packages in the '$2/All' directory"
else
  echo "Directory '$2/All' is currently empty"
fi


# [3]
# Разбор входного файла.
# Разделение списка приложений на два: обнаруженные/не обнаруженные в refRepo

# Приложения из списка желаемых, которые не удалось обнаружить в refRepo
fAppNotFound=notApp
> $fAppNotFound

cntApp=0
cntAppFound=0
cntAppNotFound=0

f=1

# id желаемых приложений, обнаруженных в refRepo, перечисленных через запятую
l=

for app in `sed 's/^[[:space:]]*//;/^$/d;/^#/d' $1`
do
  let cntApp=$cntApp+1 > /dev/null

  p=`sed -En "s%^([0-9]*)\|($app\|.*$|.*\|$app$)%\1%p" $fPkgInRefRepoAll`

  if [ -z $p ]
  then
    let cntAppNotFound=$cntAppNotFound+1 > /dev/null
    echo $app >> $fAppNotFound
  else
    let cntAppFound=$cntAppFound+1 > /dev/null
    if [ $f -eq 1 ]
    then
      l=$p
      f=0
    else
      l=$l\,$p
    fi
  fi
done

rm $fPkgInRefRepoAll

printf "\n%s\n%s" "Of the $cntApp specified applications, $cntAppFound were found and $cntAppNotFound were not
found (Look at the list in the file '$fAppNotFound')." "Continue? [y/N]:"
read ch
case $ch in
y)
  ;;
*)
  printf "\n%s\n\n" "Execution stopped."
  rm $fPkgInMinRepo
  exit
;;
esac

# [4]
# Получение списка пакетов из refRepo, которые потребуются при установке
# желаемых приложений (с учетом зависимостей)

fPkgInRefRepo=pkgInRefRepo

# SQL-запрос к базе данных refRepo

printf "%s\n%s\n%s\n%s\n%s\n%s\n%s" \
".open /var/db/pkg/repo-${repo}.sqlite" \
"with recursive myCTE(name) as (" \
  "select name from packages where id in ($l)" \
  "union" \
  "select deps.name from deps,myCTE where package_id in (select id from packages where name=myCTE.name)" \
")" \
"select packages.name,packages.version,cksum from packages,myCTE where packages.name=myCTE.name;" | \
pkg shell | sed -n 's/^\(.*\)|\(.*\)|\(.*\)/\3 \1-\2.txz/p' | sort -o $fPkgInRefRepo

# [5]
# Сравнение списков файлов (из refRepo и из minRepo)
```

```
# Список существующих, но ненужных файлов, которые надо удалить из minRepo
fDeleteList=deleteList

# Построение списка удаления
comm -13 $fPkgInRefRepo $fPkgInMinRepo | sed -n 's/^.* \(.*txz\)/\1/p' > $fDeleteList

# Список файлов для загрузки из refRepo (может потребоваться при загрузке
# файлов при помощи внешних программ, например, aria2)
#fFetchList=fetchList
# Построение списка загрузки
#comm -23 $fPkgInRefRepo $fPkgInMinRepo | sed -n 's/^.* \(.*txz\)/\1/p' > $fFetchList

rm $fPkgInMinRepo
rm $fPkgInRefRepo

# Удаление лишних файлов
printf "%s %s %s" "Warning:" `cat $fDeleteList | wc -l` "files will be deleted (see '$fDeleteList'). Continue? [y/
N]:"
read ch
case $ch in
y)
  cat $fDeleteList | xargs printf "$2/All/%s\n" | xargs rm
  ;;
*)
  printf "\n%s\n\n" "Execution stopped."
  exit
  ;;
esac

# Загрузка недостающих файлов
printf "%s\n%s" ".open /var/db/pkg/repo-${repo}.sqlite" "select name from packages where id in ($l);" | pkg shell
| xargs pkg fetch -r $repo -U -y -d -o $2

rm $fDeleteList
#rm $fFetchList


# [6]
# Построение minRepo

printf "\n%s" "Create repository located in '$2'? [y/N]:"
read ch
case $ch in
y)
  pkg repo $2
  ;;
*)
  printf "\n%s\n\n" "Execution stopped."
  exit
  ;;
esac
```

Скрипт usbdrive

```
#!/bin/sh

# Name: usbdrive
# Date: 20.03.2017
# Desc: Auto mount/umount usb flash drive

# Рабочий каталог
workdir=/media

# Файл со списком серийных номеров usb-накопителей, которые
# когда-либо подключались к системе. Файл находится в $workdir
file=.usb_device.lst

case $1 in

# Монтирование при подключении устройства
mount)
    if [ `who | wc -l` -gt 0 ]
    then
        # Имя устройства (подключенного usb-накопителя),
        # предоставленное системой (например, da0) ...
        name=`dmesg -a | grep '^da[0-9] at umass-sim[0-9]' | tail -n1 | cut -d ' ' -f1`

        # ... и его серийный номер
```

```
        serial=`camcontrol inquiry $name -S`

        # Поиск псевдонима по серийному номеру в файле $file.
        # Если usb-накопитель подключается впервые, то его
        # серийный номер записывается в $file автоматически,
        # а псевдоним можно добавить позже вручную
        grep "^[[:blank:]]*$serial\b" $workdir/$file > /dev/null
        if [ $? -eq 1 ]
        then
            echo "$serial" >> $workdir/$file
        fi

        pseu=`sed -n 's/^[[:blank:]]*'$serial'[^0-9A-Za-z#]\{1,\}\([[:alnum:]_]\{1,\}\).*/\1/p' $workdir/$file`

        # Если в файле псевдоним не указан, то вместо него будет
        # использоваться одна из латинских букв, в зависимости от
        # очередности подключения usb-накопителя (например, A)
        if [ -z $pseu ]
        then
            pseu=`echo $name | tr -d 'da' | tr '0123456789' 'ABCDEFGHIJ'`
        fi

        # Список разделов на подключенном usb-накопителе
        # (например, da0s1)
        parts=`ls /dev/ | grep ^$name's[1-9]'`

        # Признак того, что на накопителе более одного раздела
        more_slice=false

        for part in $parts
        do
            # Номер раздела
            slice=`echo $part | cut -d 's' -f2`

            if $more_slice
            then
                lnk=$pseu$slice
            else
                lnk=$pseu
                more_slice=true
            fi

            # Создание каталога для монтирования
            mkdir $workdir/$part

            # Изменение прав доступа к каталогу
            chgrp operator $workdir/$part
            chmod 770 $workdir/$part

            # Монтирование раздела (msdosfs, sync)
            /sbin/mount_msdosfs -o sync -L ru_RU.UTF-8 /dev/$part $workdir/$part

            # Если не смонтируется ...
            if [ $? -ne 0 ]
            then
                # ... то удалить созданный каталог
                rmdir $workdir/$part
            else
                # ... иначе создать ссылку на смонтированный раздел
                # (название соответствует псевдониму)
                ln -s $workdir/$part $workdir/$lnk
            fi
        done
    fi
;;

# Размонтирование при отключении устройства
umount)
    # Имя отключаемого устройства (например, da0)
    name=`dmesg -a | grep '^da[0-9]: .* detached$' | tail -n1 | cut -d ':' -f1`

    # Список каталогов (точек монтирования), оставшихся не у дел
    parts=`ls $workdir/ | grep ^$name's[1-9]'`

    for part in $parts
    do
        # Размонтирование раздела
        /sbin/umount -f $workdir/$part

        # Поиск и удаление ссылок на каталог, в который
        # монтировался раздел
        find $workdir/ -lname $workdir/$part -delete
```

```
        # Удаление каталога, в который монтировался раздел
        rmdir $workdir/$part
    done
;;

# Размонтирование при выключении компьютера
*)
    # Список всех устройств, смонтированных в $workdir
    # (например, da0s1, da0s2, ...)
    devices=`ls $workdir/ | grep '^da[0-9]'`

    for device in $devices
    do
        # Размонтирование раздела
        /sbin/umount -f $workdir/$device

        # Поиск и удаление ссылок на каталог, в который
        # монтировался раздел
        find $workdir/ -lname $workdir/$device -delete

        # Удаление каталога, в который монтировался раздел
        rmdir $workdir/$device
    done
;;
esac
```

Скрипт usb_mount_clean.sh

```
#!/bin/sh

case "$1" in
start)
        echo USB device auto mount is ready.
        /usr/local/etc/usbdrive
        ;;
stop)
        echo USB device auto umount.
        /usr/local/etc/usbdrive
        ;;
*)
        ;;
esac

exit 0
```

Скрипт wpwc

```
#!/bin/sh

# Name: wpwc / Wallpaper with calendar
# Date: 29.07.2019
# Desc: Нанесение календаря на существующее изображение
#
# Примечание:
#       Наносятся три месяца: предыдущий, текущий и последующий
#       Чтобы сделать календарь аккуратным, с правильным выравниванием чисел,
#       отсчет координат ведется от правого края в сторону левого края
#       т.е. справа налево

if [ $# -lt 1 ]
then
        printf "\nUsage:\n       \033[1mwpwc \033[0;4minput_file\033[m [ \033[4moutput_file\033[m ]\n\n"
        exit 1
fi

if [ ! -f $1 ]
then
        printf "\nFile \033[1m$1\033[m does not exists...\n\n"
        exit 2
fi

# Файл, содержащий вывод команды 'ncal -3'
textFile='ncal.temporary'
```

```
# Файл, содержащий изображение календаря (временный)
imgFile='calendar.png'

# Файл, содержащий конечный результат
if [ ! -f $2 ]
then
        outFile=$2
else
        outFile=`dirname $1`/wpwc.jpg

        if [ -f $outFile ]
        then
                fDate=`stat -f%m $outFile`
                cDate=`date -j 0000 +%s`
                if [ $fDate -gt $cDate ]
                then
                        printf "\nCome back tomorrow\n\n"
                        exit 3
                fi
        fi
fi

# Шрифт для текста
# Посмотреть названия шрифтов: convert -list font
#font='PT-Sans'
font='Pfennig'

# Размер шрифта
pSize=14

# Основной цвет
fgColor=gray95

# Цвет тени
bgColor=gray20

# Цвет для выделения
mkColor=YellowGreen

# Расстояние между строками или высота поля, в которое вписывается
# календарное число
dy=$(($pSize + 6))

# Ширина поля, в которое вписывается календарное число
dx=$(($dy + $dy / 4))

# Расстояние между отдельными месяцами
sp=10

# Смещение от края
x0=10
y0=50

# Текущий день
now=`date "+%e"`

# Координаты 'now' в календаре (для выделения цветом)
pNow=

ncal -3 > $textFile

year=`head -n 1 $textFile | cut -w -f 5`
pMonth=`head -n 1 $textFile | cut -w -f 2`
cMonth=`head -n 1 $textFile | cut -w -f 4`
nMonth=`head -n 1 $textFile | cut -w -f 6`

for n in `seq 2 8`
do
        y=$(($y0 + $n * $dy + $dy / 4))

        line=`sed -n "${n}s/ /./g p" $textFile`
        k=18
        for v in `seq 1 3 55`
        do
                a=`echo $line | cut -c $v-$(($v + 1)) | tr '.' ' '`
                case $v in
                1)
                        x=$(($x0 + 3 * $sp + k * $dx))
                        ;;
                4|7|10|13|16|19)
                        x=$(($x0 + 2 * $sp + k * $dx))
```

```
                ;;
        22|25|28|31|34|37)
                x=$(($x0 + 1 * $sp + k * $dx))
                if [ "$a" = "$now" ]
                then
                        pNow=+${x}+${y}
                        pNowX=$x
                        pNowY=$y
                fi
                ;;
        40|43|46|49|52|55)
                x=$(($x0 + k * $dx))
                ;;
        *);;
        esac
        p=+${x}+${y}
        k=$(($k - 1))
        if [ $a ] && !([ "$a" = "$now" ] && [ $v -ge 22 ] && [ $v -le 37 ])
        then
                fText="${fText} -annotate ${p} ${a}"
        fi
    done
done

width=$(($x0 + 3 * $sp + 20 * $dx))
height=$(($y0 + $y + $dy))

rect="$(($width - $pNowX + $dx / 10)),$pNowY $(($width - $pNowX - $dx + $dx / 10)),$((pNowY + $dy))"

convert -size ${width}x${height} xc:none \
        -font ${font} -pointsize ${pSize} \
        -gravity NorthEast \
        -stroke ${bgColor} -strokewidth 4 \
        -annotate +${x0}+${y0} $year \
        -annotate +$(($x0 + 2 * $sp + 14 * $dx))+$(($y0 + $dy)) $pMonth \
        -annotate +$(($x0 + $sp + 8 * $dx))+$(($y0 + $dy)) $cMonth \
        -annotate +$(($x0 + 2 * $dx))+$(($y0 + $dy)) $nMonth \
        -annotate $pNow $now \
        $fText \
        -annotate $pNow $now \
        -blur 0x4 \
        -fill ${fgColor} -stroke none \
        -annotate +${x0}+${y0} $year \
        -annotate +$(($x0 + 2 * $sp + 14 * $dx))+$(($y0 + $dy)) $pMonth \
        -annotate +$(($x0 + $sp + 8 * $dx))+$(($y0 + $dy)) $cMonth \
        -annotate +$(($x0 + 2 * $dx))+$(($y0 + $dy)) $nMonth \
        $fText \
        -fill ${mkColor} \
        -draw "fill-opacity 0.4 rectangle $rect" \
        -fill ${fgColor} -stroke none -strokewidth 4 \
        -annotate $pNow $now \
${imgFile}
# -fill ${mkColor} -stroke ${fgColor} -strokewidth 1

composite -gravity SouthEast -compose ATop ${imgFile} "$1" "${outFile}"
rm $textFile
rm $imgFile
```