

Журнал «Linux user from the hadean eon» №000000

saahriktu

16 ноября 2017 г.

Обратите внимание, что на x86/x86_64 в ядерной консоли без иксов по дефолту SDL2 собирается исключительно с dummy видеодрайвером. При том, что существует возможность включить поддержку того же directfb, но это делается принудительно вручную через опцию --enable-video-directfb.
Если нужно задать грет'у выражение, которое содержит символ табуляции '\t' (это может быть полезно, например, для обработки вывода du), то можно использовать формат --perl-regex (сокращённая форма ключа — «-P»).
Сборка nethack 3.6.0: cd nethack-3.6.0/ vim sys/unix/hints/linux sys/unix/setup.sh sys/unix/hints/linux make all make install
Сборка perl6: perl Configure.pl --gen-moar --gen-nqp --backends=moar --prefix=/usr && make && make install
Подружить ncurses и KOI8-R можно при помощи #include <locale.h> setlocale(LC_ALL, "ru_RU.KOI8-R"); Также следует обратить внимание на то, что addch() работает с unsigned char.
После обновления glibc'а надёжнее удалить файл /usr/lib/locale/locale-archive и сгенерировать нужные локали заново.
При сборке fuse-emulator'a с SDL следует явно указывать опцию --with-sdl.
При установке telegram-cli достаточно просто скопировать бинарник в /usr/bin, а ключ в /etc/telegram/server.pub. После чего остаётся добавить строчку «alias telegram-cli="telegram-cli -k /etc/telegram/server.pub"» в ~/.bashrc.
Обратите внимание на то, что оболочка после логина в нативную консоль отличается от эмуляторов терминала в графических средах и оболочки в виртуальных терминалах, которые были открыты через openvt, тем, что последние чатают ~/.bash_profile, игнорируя ~/.bashrc.
Файл /etc/udev/rules.d/60-persistent-storage.rules является устаревшим, и может приводить к ошибке «udev[XXXX]: failed to execute '/lib/udev/socket:/org/kernel/udev/monitor' 'socket:/org/kernel/udev/monitor': No such file or directory»
Патчи для сборки/сборка splint: «LDFLAGS = -lfl» — src/Makefile sparse 0.5.0: «LDFLAGS += -g -lpthread -lcurses -ldl» — Makefile lapack-3.6.0: cmake -DCMAKE_INSTALL_PREFIX=/usr -DBUILD_SHARED_LIBS=ON CMakeLists.txt
В bash'e и readline можно отключить поддержку юникода: sed -i 's/^opt_multibyte=yes/opt_multibyte=no/' configure
В Ruby поддержка локали KOI8-R включается так: #!/usr/bin/ruby # encoding: KOI8-R
Из библиотеки glibc, которая обеспечивает основную поддержку локали KOI8-R, выпилить поддержку юникода нельзя. Особенно это касается функций strftime(), printf(), vfprintf(), strtod(), strtold(), vfscanf(), confstr(), regcomp(), regexec(), fnmatch() и error(). Функции продолжают работать с однобайтными кодировками, но могут местами жрать больше оперативки. Поэтому лучше использовать, например, функции putchar() и putstr() вместо printf() где это только можно. Кстати, на более низком уровне и putchar() и printf() функции вызывают _IO_putc_unlocked() (а для юникода printf вызывают _IO_putwc_unlocked()). Собственно, сначала происходит блокировка stdout через _IO_acquire_lock(), которая снимается через _IO_release_lock(). Это уровень подсистемы libio из состава glibc. Собственно, _IO_putc_unlocked() является макросом, который дёргает _IO_BE(), _IO_write_ptr, _IO_write_end, __overflow(). puts() же является алиасом _IO_puts(). _IO_puts() является полноценной функцией, которая помимо блокировок дёргает _IO_vtable_offset(), _IO_fwide(), _IO_sputn() и MIN(). Попытавшийся распутать это дальше обнаружит то, что за пределы glibc'a все эти клубки алиасов и макросов не ведут.

Не все функции в glibc определены явно. В glibc 2.24 появились, например, новые функции, которые требуют доопределения перед их явным использованием: `#include <tgmath.h>`

```
double nextup(double);
float nextupf(float);
long double nextupl(long double);
double nextdown(double);
float nextdownf(float);
long double nextdownl(long double);
```

Без доопределения функции, вопреки официальной документации, возвращают `int`, а компилятор выдаёт предупреждения.

The 'nextup' function returns the next representable neighbor of X in the direction of positive infinity. If X is the smallest negative subnormal number in the type of X the function returns '-0'. If X = '0' the function returns the smallest positive subnormal number in the type of X. If X is NaN, NaN is returned. If X is +inf, +inf is returned. 'nextup' is based on TS 18661 and currently enabled as a GNU extension. 'nextup' never raises an exception except for signaling NaNs.

The 'nextdown' function returns the next representable neighbor of X in the direction of negative infinity. If X is the smallest positive subnormal number in the type of X the function returns '+0'. If X = '0' the function returns the smallest negative subnormal number in the type of X. If X is NaN, NaN is returned. If X is -inf, -inf is returned. 'nextdown' is based on TS 18661 and currently enabled as a GNU extension. 'nextdown' never raises an exception except for signaling NaNs.

В Perl'e 5 «use bignum;» увеличивает точность вычислений, поскольку задействует работу с большими числами. По дефолту точность определяется автоматически (или ограничивается примерно 42-43), но может быть уточнена через опции модуля (например, «use bignum a => 128;»). Последние цифры в таких вычислениях являются неточными, поэтому ручную точность лучше устанавливать с запасом. В Perl 6 нет модуля bignum, но большие числа задействованы по дефолту.

В Perl 5 конкатенация строк происходит через символ '.', а в Perl 6 через символ '~'. В PHP echo для конкатенации использует как и Python print символ ','. print в Python 3 в отличие от Python 2 требует формы print(). puts в Ruby для конкатенации строк использует символы '+' и ',' (в последнем случае происходит переход на новую строку), но при этом никакие вычисления на лету происходить не могут — переменные с числовыми значениями должны быть подставлены в форме var.to_s().

В Perl 6 есть функция atan(), а в Perl 5 её нет. Вместо неё в Perl 5 функция atan2(Y,X), которая возвращает арктангенс Y/X (угол между OX и OA, где O (0; 0), а A (X; Y).

В Python 2 и Ruby, как и в C, «7.0» — число с плавающей запятой, а «7» — целое. Таким образом, например, 77.0 / 17.0 = 4.529412; 77 / 17 = 4.

В PHP 7, Python 3 и Perl 5/6 не так: 77 / 17 = 4.529412.

Пример работы с джойстиком через ядро Linux напрямую:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <linux/joystick.h>
#include <math.h>

int main(){
    int i;
    int jstck0 = open ("/dev/input/js0", O_RDONLY);
    struct JS_DATA_TYPE jsdt;
    for (;;) {
        if (read (jstck0, &jsdt, JS_RETURN) != JS_RETURN) {
            /* error */
        }
    }
}
```

```

        printf("Something went wrong...\n");
        break;
    }
    for (i=0; i<13; i++) if (jsdt.buttons & (int)pow(2, i))
        printf("Button %d pressed.\n", i);
    printf("X = %d, Y = %d.\n", jsdt.x, jsdt.y);
    sleep(1);
}
close(jstck0);
}

```

Таким образом можно добавлять поддержку джойстика в обычные игры на ncurses без необходимости задеирования дополнительных библиотек.

<p>Изменить размер шрифта в Pidgin'e можно через создание собственного стиля в файле <code>~/.purple/gtkrc-2.0</code>. Например, так:</p> <pre> style "saahriktu-style" { font_name = "Sans 15" } widget_class "*" style "saahriktu-style" </pre>
<p>В Firefox'e отключить предупреждения о передаче данных форм не через HTTPS можно следующим способом: в <code>about:config</code> достаточно выставить значение переменной <code>security.insecure_field_warning.contextual.enabled</code> в <code>false</code>.</p> <p>А для отключения значка серого замка, который перечёркнут красной линией, можно выставить в <code>about:config</code> значение переменной <code>security.insecure_password.ui.enabled</code> в <code>false</code>.</p>
<p>Получить список имён установленных пакетов в Slackware можно командой <code>«ls /var/log/packages/ rev cut -d - -f 4- rev»</code>.</p> <p>А получить список установленных пакетов с одинаковыми именами (в Slackware такое возможно если активно красноглазить) можно командой <code>«ls /var/log/packages/ rev cut -d - -f 4- rev uniq -d»</code>.</p>
<p>При выполнении команд вида <code>«yes n find dir1 -type f -exec mv -i "{}" dir2 \;»</code> не следует забывать <code>-i</code>, иначе конструкция потеряет половину смысла, и файлы с одинаковыми именами будут перезаписаны друг другом.</p>
<p>Наверное, не так уж многие знают, что существует зеркало исходников для слакбилдов со <code>slackbuilds.org</code>. Создать/синхронизировать локальное зеркало можно такой командой как <code>«rsync -rtLvH --delete-after --delay-updates --safe-links --copy-links --ignore-errors --ignore-existing rsync://slackware.uk/sbosrcarch/by-name/ /mnt/mpt0/system/slackware/sbosrcarch/»</code> (вместо <code>«/mnt/mpt0/system/slackware/sbosrcarch/»</code> подставить путь к своей директории).</p> <p>Сами слакбилды со <code>slackbuilds.org</code> можно тоже синхронизировать через github. Их зеркало на github'e можно найти, например, здесь: https://github.com/willysr/slackbuilds.</p>
<p>Улучшаем поддержку UTF-8 в Slackware.</p> <p>Для того, чтобы <code>man</code> поддерживал локаль <code>ru_RU.UTF-8</code>, пакет нужно пересобрать с этим слакбилдом: https://github.com/saahriktu/modified-slackbuilds/tree/master/man, а затем в <code>/etc/man.conf</code> прописать <code>«NROFF /usr/bin/groff -Dutf8 -Tutf8 -mandoc»</code>.</p> <p>Из коробки <code>/usr/bin/vi</code> является симлинком на текстовый редактор <code>elvis</code> в котором нет нормальной поддержки UTF-8 (всё кракозябрами). А к <code>vi</code> может обращаться тот же <code>git</code>. Простейший вариант устранения проблемы: <code>«removepkg elvis && ln -s /usr/bin/vim /usr/bin/vi»</code>. Ещё можно, например, поставить или <code>ex-vi</code> (устанавливается в <code>/opt/ex-vi/bin/vi</code>) или <code>nvi</code>, и прописать симлинк на установленный редактор.</p>
<p>Если команда <code>eject</code> отказывается работать от обычного юзера с ошибкой <code>«eject: unable to eject, last error: Неприменимый к данному устройству ioctl»</code>, но работает от <code>root'a</code>, то может помочь команда от <code>root'a</code> <code>«chmod 4755 /usr/bin/eject»</code>.</p> <p>Если понадобится вернуть всё как было, то поможет команда от <code>root'a</code> <code>«chmod 0660 /usr/bin/eject»</code>.</p>

Если при обновлении Debian'a происходит ошибка проверки хэшей, то может помочь удаление файла `dists/stable/InRelease`.

В Firefox'е принудительно включить многопроцессорность в `about:config` с 47-й версии можно так:

```
browser.tabs.remote.autostart*=true  
browser.tabs.remote.force-enable=true // создать параметр  
dom.ipc.processCount=20
```

Статус мультипроцессности можно проверять в `about:support`, строчка «Multiprocess Windows». При включенной мультипроцессности там будет нечто наподобие «1/1 (Enabled by user)». Ключевое слово «Enabled».