



Заметкоделие в Emacs

с использованием пакета howm



ОГЛАВЛЕНИЕ

Вступление	2
История создания <code>howm</code>	3
Клавиатурные сочетания в Emacs	5
Установка пакета <code>howm</code>	8
Основное меню <code>howm</code>	12
Создание заметок	16
Работа со списками	19
Сортировка заметок в списках	26
Работа со ссылками	29
Ключевые слова в <code>howm</code>	35
Функции поиска	41
Функция фильтрации	44
Удаление заметок	47
Сшивание заметок	49
Работа с длинными заметками	52
Работа с буферами	56
Быстрые заметки	58
Дублирование	60
Произвольные названия	61
Система планирования в <code>howm</code>	62
Управление задачами	70
Короткие команды в списках	77
Дополнительные настройки	78
Вопросы и опасения	86
Прил.№1. Список основных команд	87
Прил.№2. Оч.кр.руководство	89

ВСТУПЛЕНИЕ

Перед вами пособие по ведению заметок в Emacs с помощью пакета `howt`. Я исхожу из допущения, что читатели уже осознали важность «мышления письмом» и знакомы с основными принципами управления знаниями.

Собственно, основных принципов по пальцам пересчитать:

1. Делайте заметки обо всем, что вы считаете полезным, важным, интересным.
2. Ограничивайте объем заметки одной идеей. Так их легче будет потом комбинировать.
3. Линкуйте заметки, идеи которых вступают в диалог.
4. Используйте надежную программу, которой можно доверить свои тексты.

О такой программе пойдет речь в данном пособии. Я постараюсь показать, почему функции из арсенала `howt` важны для организации интеллектуального труда. Однако у меня нет задачи навязать читателю именно `howt`. Существуют и другие хорошие решения (даже в рамках Emacs). Я лишь надеюсь, что вы по достоинству оцените простоту и надежность этого пакета.

Текст рассчитан на абсолютных новичков в Emacs, только приступивших к изучению основ. По ходу повествования я буду пояснять некоторые универсальные приёмы, применимые не только в `howt`. Разумеется, людям опытным все это может наскучить. Они вольны выбирать разделы пособия по собственному усмотрению.

Я не стремился создать исчерпывающее техническое руководство, поэтому какие-то частности остались за скобками. Моя задача в том, чтобы продемонстрировать возможности `howt`, как инструмента для ведения картотеки. И, разумеется, это мое субъективное восприятие системы. Пособие рассказывает не про то как нужно, а лишь про то как можно использовать `howt`.

В процессе написания этого текста я неоднократно донимал вопросами создателя пакета `howt` – Казуюки Хираока. Хочется выразить ему благодарность за терпение и желание помочь. Домо аригато!

Текст распространяется на основе GNU Free Documentation License и доступен для свободного использования.

Суховский Андрей, 2024

ИСТОРИЯ СОЗДАНИЯ HOWM

Название `howm` – это акроним японской фразы или лозунга *Hitori Otegaru Wiki Modoki* (一人お手軽 Wiki もどき). Что можно перевести как – «Удобная персональная вики-подобная система».

Пакет `howm` удивляет компактностью и элегантностью решений. В данном случае характерно, что это сделано в Японии. Автор – Казуюки Хираока – является профессором в Asia University на кафедре Науки о данных. Проекту уже больше 20 лет. Хираока разработал его в 2002 году.

Чем же подкупает `howm`? Что в нем такого особенного? Вкратце – он придуман для неорганизованных людей.

Как отмечает Хираока, «Если заметки не организовывать, их будет трудно читать, а если организовывать – сложно писать. Проблема в достижении компромисса. “Хлопотность” – *Оккю* (おくう) – ключевое слово. Между “я могу это сделать” и “я могу сделать это легко” есть большая разница... Тут многое зависит от человека, но я придаю большее значение легкости письма. Я разработал этот заметочник, чтобы сформировать среду, в которой можно легко вести записи, не испытывая давления “организованности” и при этом не обломаться».

Хираока пишет, что жесткая структура и деление на категории быстро становятся помехой. Девиз `howm` – «Write fragmentarily and read collectively» – на мой взгляд, хорошо отражает саму суть картотеки.

`Howm` подходит для работы с любыми текстовыми файлами¹. Неважно будет это `txt`, `org` или `markdown`. Разместив файлы в папке `howm`, вы делаете их частью базы знаний.

Каждая новая заметка в `howm` – это новый файл, который получает уникальный номер, основанный на дате и времени создания. Ни тегов, ни даже ссылок в традиционном понимании нет.

Для организации работы с заметками Казуюки Хираока прибегнул к одному хитрому трюку. В пакете `howm` ссылка – это триггер поискового запроса. Итогом является список, в которой указаны все заметки, где фигурирует ссылка (имя файла, заголовок и т.п.)².

Функционал `howm` был вдохновлен системой организации данных японского экономиста Юкио Ногучи.

¹ Поскольку `howm` – это побочный режим (*minor mode*) в *Emacs*, что позволяет ему дружить с другими режимами.

² Похожим образом, кстати, работа со ссылками была организована в *Notational Velocity*, приложении, которое разработал Захарий Шнейров (Zachary Schneirov) в 2002 году.

Вкратце система Ногучи предлагает заменить тематическое распределение информации распределением на основе частотности обращений. Таким образом, наиболее важные документы всегда оказываются под рукой.

Howmt следует схожему принципу. Категории и разделы отсутствуют³. Вместо этого общий список, в котором заметки распределены по дате последней правки. Такой алгоритм работы с информацией на поверку оказывается гораздо эффективней, чем попытки навести идеальный порядок.

Архитектура howmt была вдохновлена разными приложениями. Среди них Q-Pocket, ChangeLogMemo, HyperCard, HashedWiki и др.

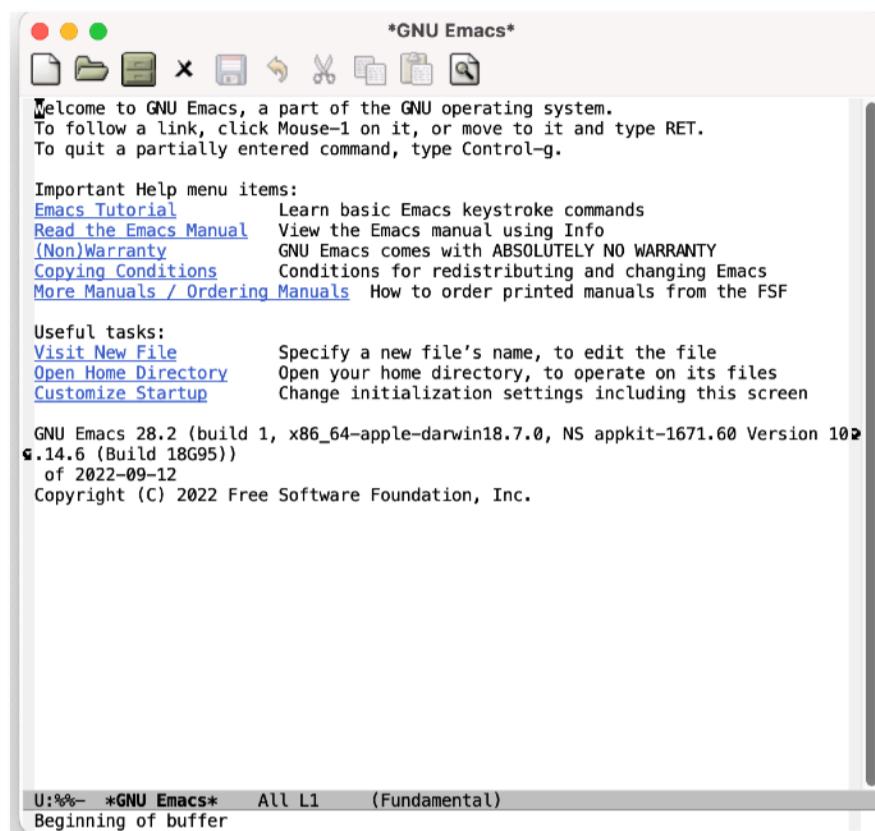
Синтаксис howmt может показаться необычным при первом соприкосновении. Сегодня публика привыкла к формату markdown и желает видеть его повсюду. В оправдание howmt можно сказать, что он появился раньше.

Мы перейдем к обсуждению возможностей howmt на стр. 12, но перед этим мне необходимо сделать пару вводных замечаний об использовании Emacs, установке новых пакетов и т.п.

³ Данный подход также распространен среди адептов zettelkästen.

КЛАВИАТУРНЫЕ СОЧЕТАНИЯ В EMACS

Я исхожу из того, что вы уже установили Emacs на свой компьютер.



Поздравляю! Вы можете работать с документами, создавать и редактировать заметки, отслеживать разные задачи и делать множество других вещей.

В следующем разделе мы поговорим об установке пакета `howm`, но сначала я должен рассказать о клавиатурных сочетаниях.

Если вы только приступили к знакомству с Emacs, очень многие вещи покажутся вам необычными. Лучшее, что можно посоветовать, – это пройти встроенный самоучитель, который так и называется – `Emacs Tutorial`. Он поджидает вас на начальном экране, сразу после запуска программы (первая гиперссылка).

У Emacs очень длинная история и свои, сформировавшиеся за десятилетия, термины, способы решать проблемы и т.п.

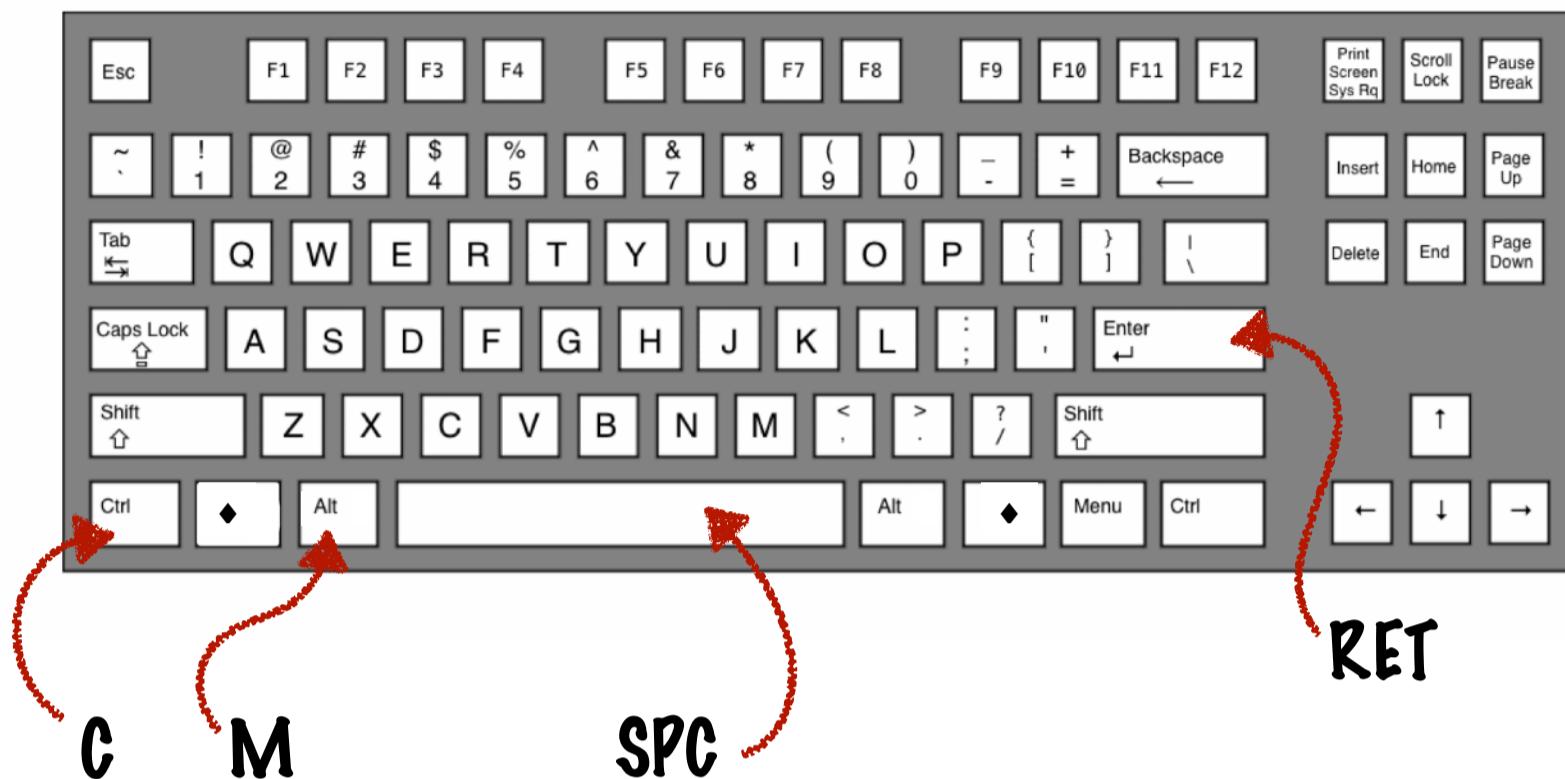
Одной из важных особенностей Emacs является «клавиатуроцентричность». Абсолютно любое действие можно совершить, не покидая своими пальцами клавиатуры. «Долой мышиную возню с текстом» – таким мог бы быть один из лозунгов Emacs.

Кроме общих для Emacs горячих клавиш, существуют специфические команды под разные программные пакеты.

В следующих разделах пособия мы рассмотрим команды, используемые в `howm`⁴.

⁴ Полный список клавиатурных сочетаний `howm` можно посмотреть в Приложении №1.

Задача этой главы лишь в том, чтобы показать, как расшифровываются принятые в Emacs обозначения.



Команда **C-x C-s** означает такую последовательность действий:

1. Жмем **Control** и, не отпуская (поэтому там дефис), жмем **x**
2. Жмем **Control** и, не отпуская, жмем **s**

На самом деле, это действие можно упростить, так как у нас обе части команды начинаются с **Control**.

Нажимаем **Control** и, удерживая, нажимает сначала **x**, а потом **s**.

Если в клавиатурном сочетании указана заглавная буква, именно ее и нужно использовать (следовательно, задействуется еще и **Shift**).

Рассмотрим для примера команду **C-u C-c , K**

Она означает, что нужно нажать **Control** и, удерживая, нажать **u**. Дальше, все еще удерживая **Control**, нажать **c**. После этого нажать на запятую и заглавную **K**⁵.

Команда **M-x** означает:

Жмем **Alt** и, не отпуская, жмем **x**

Почему **M-x**? Буква **M** - сокращение от **Meta**. Была раньше такая клавиша на некоторых клавиатурах. Сегодня вместо нее чаще всего используется клавиша **Alt**, а традиция обозначать ее буквой **M** осталась.

⁵ Не пугайтесь, большинство команд в Emacs состоит из 2-4 символов.

Эта команда, кстати, является вызовом специального диалога – минибуфера внизу экрана.

Клавиатурное сочетание – это просто способ инициировать определенную команду. Однако ее можно вызвать и иначе. Например, клавиатурное сочетание `C-c`, с отвечает в `howt` за создание новой заметки. *Emacs* в этом случае выполняет команду `howt-create`. Если вы вызовете минибуфер (`M-x`) и напечатаете `howt-create`, то также создадите новую заметку. В общем, клавиатурные сочетания – это способ не вбивать длинные команды вручную.

Нужно иметь в виду, что все команды рассчитаны на английскую раскладку. Если вы поменяете раскладку на русский, *Emacs* ничего не поймет.

Переключать раскладку туда-сюда в процессе набора текста – дело муторное. К счастью есть простое решение.

Команда `C-\` вызовет диалог для выбора языка клавиатурного набора. Укажите `russian-computer`. Основная раскладка клавиатуры останется при этом английской, *Emacs* будет понимать все команды, но печатать вы сможете на великом и могучем (или любом другом языке на выбор).

Последующее приключение между английским и русским также осуществляется командой `C-\`

В большинстве случаев горячие клавиши не нужно целенаправлено запоминать. В процессе использования они сами «прилипнут» к пальцам. По крайней мере, так произойдет с ходовыми, повседневными командами. Моторная память вырабатывается довольно быстро. Вы как будто изучаете аккорды для игры на гитаре.

Конечно, радикально иной подход к взаимодействию с интерфейсом может обескураживать. Навигация в GUI программах удобнее тем, что она интуитивно понятна. На худой конец, можно методом тыка понять, за что отвечает определенная кнопка. В *Emacs* интуитивности меньше... Но удобство величина относительная. На короткой дистанции графический интерфейс выручает, на длинной – становится обузой.

Это любого приложения касается. Понятно, что и Microsoft Word лучше копировать текст командой `Ctrl-c` и вставлять с помощью `Ctrl-v`, чем ползти курсором к соответствующим иконкам.

В *Emacs*, к слову, есть и иконки и приборная панель, так что никто не посягает на вашу зону комфорта. Однако, как только вы привыкнете к клавиатурным аккордам, прежний метод покажется вам удивительно неуклюжим.

УСТАНОВКА ПАКЕТА HOWM

Даже сам по себе, безо всяких усовершенствований, Emacs является – кроме всего прочего – прекрасным текстовым редактором. Множество замысловатых манипуляций с текстом доступны вам прямо после его установки.

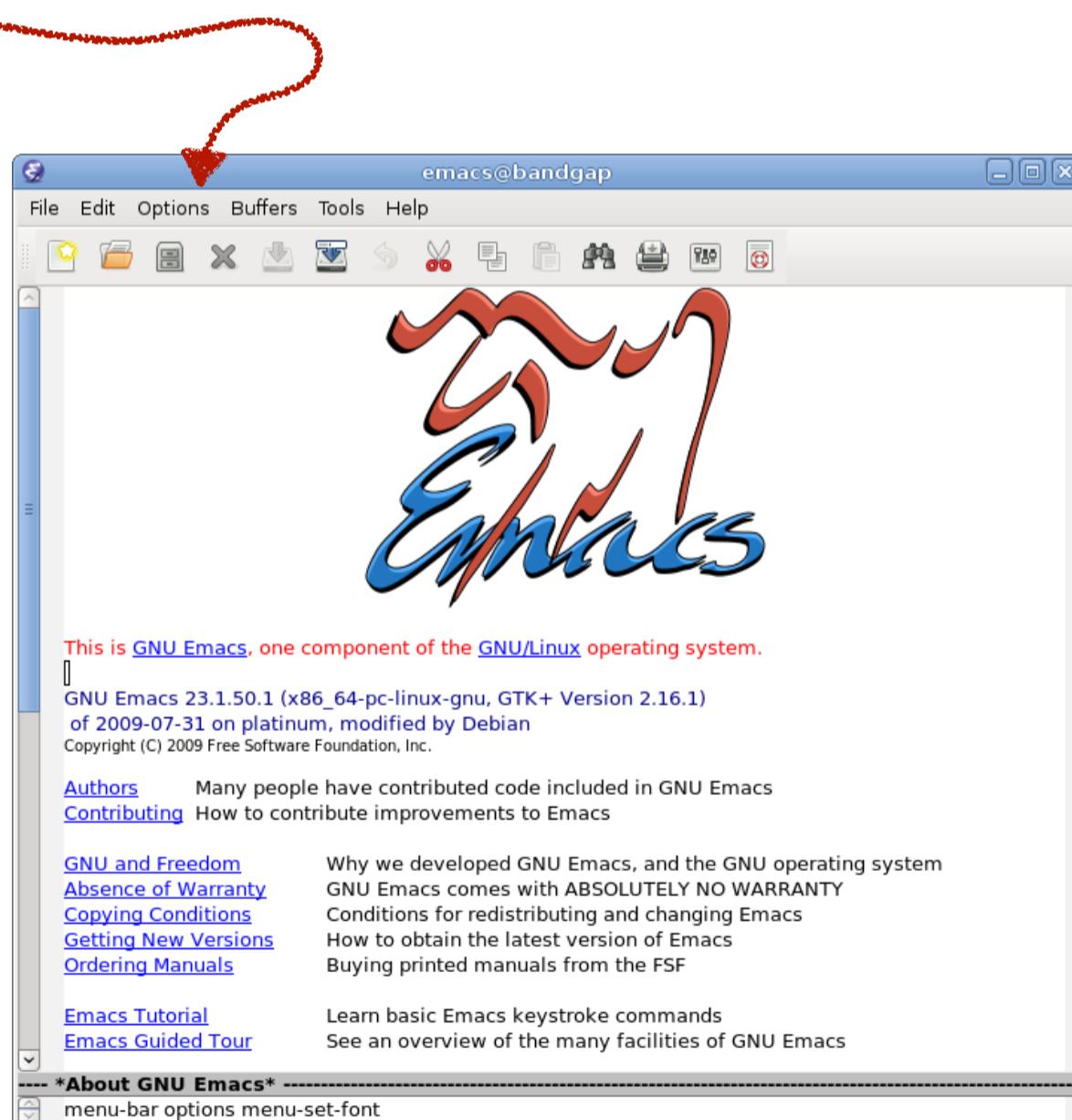
Однако возможности Emacs могут быть адаптированы под ваши нужды благодаря дополнительным пакетам.

Howm устанавливается на Emacs точно так же, как и множество других расширений. Поэтому, давайте разберем общие принципы.

Проверьте, какие пакеты уже установлены.

Чтобы посмотреть их список, зайдите во вкладку Options и выберите графу Manage Emacs Packages. Где это? Что это?

Когда вы запускаете Emacs, он открывает приветственное окно, с логотипом и вводной информацией. В шапке прямо над иконками располагается меню с разными опциями⁶.



⁶ Если у вас Macbook, дизайн будет незначительно отличаться. Меню будет на самом верху экрана.

Итак, нажимаем Options → Manage Emacs Packages. Перед вами откроется новое окно с длиннющим списком пакетов.

В самом конце будут перечислены пакеты *built-in* (они уже установлены). Остальные пакеты вы устанавливаете по своему усмотрению. Делать это стоит вдумчиво, чтобы не превратить Emacs в свалку ненужных возможностей.

Архив Elpa (аббревиатура расшифровывается как Emacs Lisp Package Archive) доступен вам по умолчанию. Здесь располагаются самые кошерные опенсорсные продукты.

Однако многие разработчики размещают свои творения в менее щепетильном архиве Melpa (Milkypostman's Emacs Lisp Package Archive).

Наша задача – подсказать Emacs путь к этому архиву, так как пакет howm находится там.

Обычно в Emacs существует много разных путей достичнуть желаемого результата.

Я покажу как подключить дополнительный архив двумя способами: сначала привычным, потом удобным.

Привычный способ для многих пользователей – использование панели инструментов.

1. Зайдите во вкладку Options
2. Выберите из списка – Customize Emacs
3. В следующем списке выберите – Top Level Customization Group
4. Найдите пункт Applications
5. В новом списке выберите пункт Package

Дальше нужно выбрать опцию Package Archives (Мы добрались до нужной настройки)

6. Нажмите на кнопку INS
7. Введите имя Melpa и адрес: <https://melpa.org/packages/>
8. Наверху страницы нажмите кнопку Apply and Save



Теперь более быстрый способ сделать тоже самое.

Большинство действий в Emacs, как я уже сказал, осуществляется с помощью разных сочетаний клавиш.

1. Нажмите **M-x**
2. Наберите в открывшемся внизу диалоге – `customize-variable` и нажмите RET.
3. Дальше наберите `package-archives` и еще раз нажмите RET.

Вас перебросит на страницу Package Archives. Затем нужно будет заполнить поля, как в предыдущем примере (там это пункт семь).

Так или иначе теперь вы подключили к своему Emacs архив Melpa с дополнительными пакетами.

Вы можете посмотреть новый расширенный список, опять прибегнув к помощи приборной панели:

`Options → Manage Emacs Packages`

Или использовать простую команду: `M-x list-packages`

В Emacs полно встроенных подсказок и упрощалок работы. Вам не нужно набирать команду до конца. Достаточно набрать часть слова и затем щелкнуть по клавише табуляции. Emacs дозаполнит команду за вас. Если есть другие похожие команды, вам предложат варианты.

Список пакетов перед вами. Он большой...

Установка и удаление пакетов в Emacs с помощью архива очень проста.

Я опять покажу сначала привычный, а потом быстрый способ.

Пакеты расположены в алфавитном порядке, так что можно просто пролистать вниз. Но это муторно. Легче прибегнуть к поиску.

Нажмите **C-s** и дальше наберите имя интересующего вас пакета (в нашем случае это `howt`). Чтобы перейти к следующему результату продолжайте нажимать **C-s**

Если щелкнуть по имени пакета, в новом окне откроется информация о нем (всегда полезное с ней ознакомиться). Там же будет кнопка установки.

Еще один способ сделать это – разместить курсор рядом с пакетом в списке и нажать клавиши `i` и `x`. Нажимая `i` вы просто помечаете пакет. Можно выбрать сразу несколько пакетов, а потом нажать `x`, чтобы Emacs их загрузил.



Теперь о самом быстром способе установки пакетов.

1. Наберите **M-x**

2. Наберите `package-install` и нажмите RET

3. Наберите имя пакета - `howm` в нашем случае - и нажмите RET

Теперь, когда пакет `howm` установлен тем или иным способом, нужно подсказать Emacs, что это не какое-то инородное тело.

1. Используйте команду C-x C-f

2. Дальше введите `.emacs` и нажмите RET

В открывшемся файле⁷ ниже имеющихся записей разместите строку:

`(require 'howm)`

Обратите внимание, что апостроф перед словом `howm` должен быть прямым.

Простой орфографический ‘ не подойдет.

3. Сохраните изменения: C-x C-s

Этих действий будет достаточно для исправной работы пакета⁸.

⁷ Файл `.emacs` хранит все модификации системы, которые вы осуществили. Если этот файл удалить, Emacs вернется к настройкам по умолчанию.

⁸ Некоторые модификации могут, тем не менее, улучшить работу системы (см. раздел «Дополнительные настройки»).

ОСНОВНОЕ МЕНЮ HOWM

С базовыми функциями `howm` можно познакомиться за считанные часы. Если вам нужен именно такой экспресс-курс, загляните в конец книги. Там будет дана краткая инструкция.

Однако я бы рекомендовал вам не спешить и более пристально изучить нюансы использования системы.

Первое и самое важное, что вам необходимо запомнить – это команда запуска `howm`: `C-c , ,`

Так вы попадаете в основное меню. Дальше двигаться намного легче. Это что-то подобное капитанскому мостику, с которого можно быстро отдавать приказы и следить за разными отсеками судна⁹.

Вся информация в окно не влезает. Пролистывать вниз можно, нажимая `Space` (или `C-v`), вверх – `Backspace` (или `M-v`)¹⁰.

Подчеркнутые элементы являются активными. Разместив на них курсор и нажав `RET`, вы выполните команду или перейдете по ссылке.

Базовое перемещение курсора в `Emacs` осуществляется клавишами `C-n` и `C-p` – вниз и вверх, `C-f` и `C-b` – вперед и назад.

В основном меню `howm` вы можете перемещаться между активными элементами с помощью клавиши табуляции:

`Tab` – скачем вперед

`Tab` вместе с `Alt` – скачем назад. В некоторых системах это сочетание уже зарезервировано. В этом случае можно использовать `C-[Tab`

Можно, конечно, перемещать курсор и с помощью мышки или тачпада, если на данном этапе это привычнее.

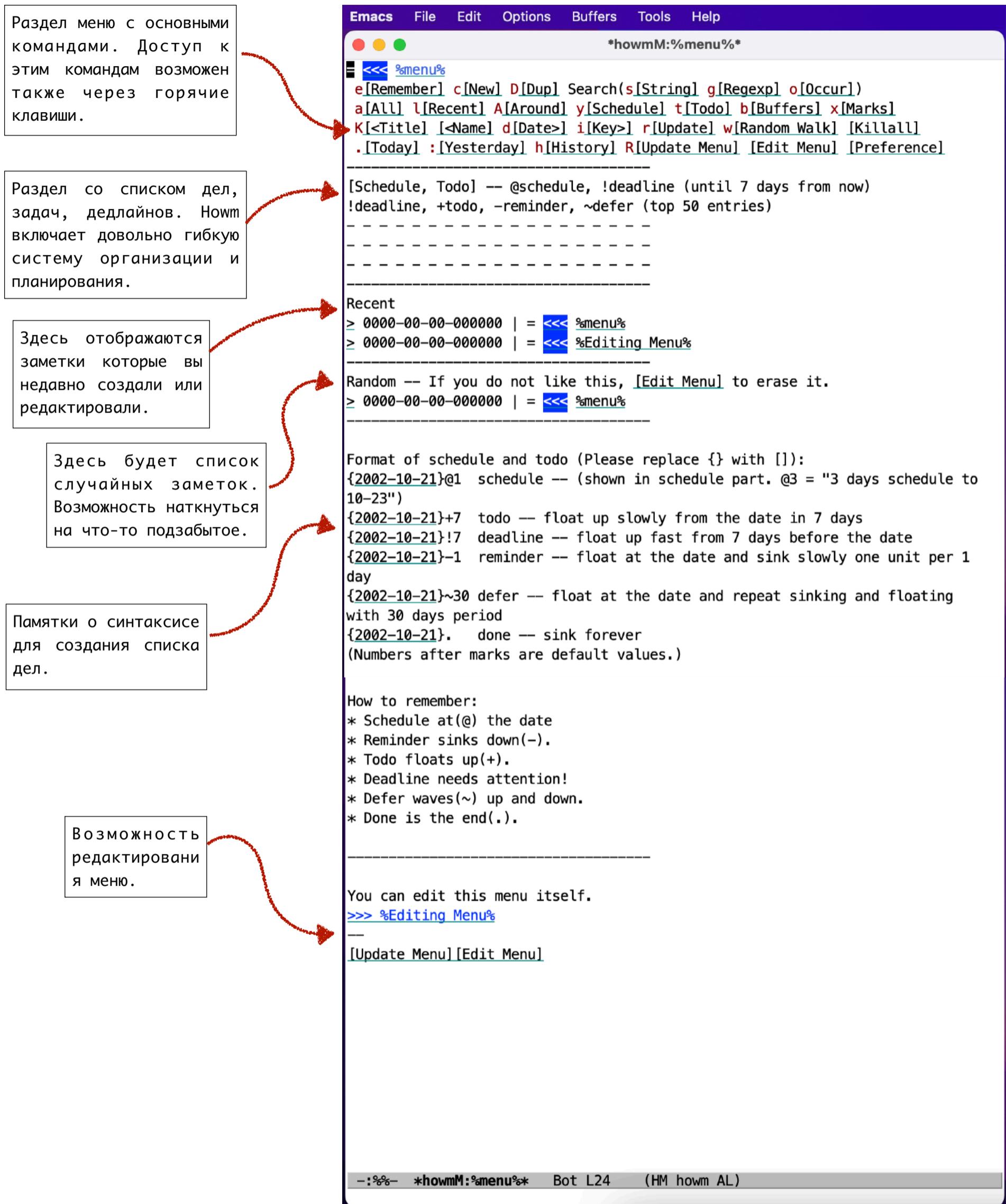
На самом деле, почти любое действие из основного меню `howm` быстрее выполнить безо всяких перемещений курсора. Для каждой команды задана клавиша (см. илл. на следующей странице). Нажав на `S`, вы инициируете поиск, нажав на `C` – создание заметки и т.д.

Для выхода из меню `howm` достаточно нажать клавишу `q`.

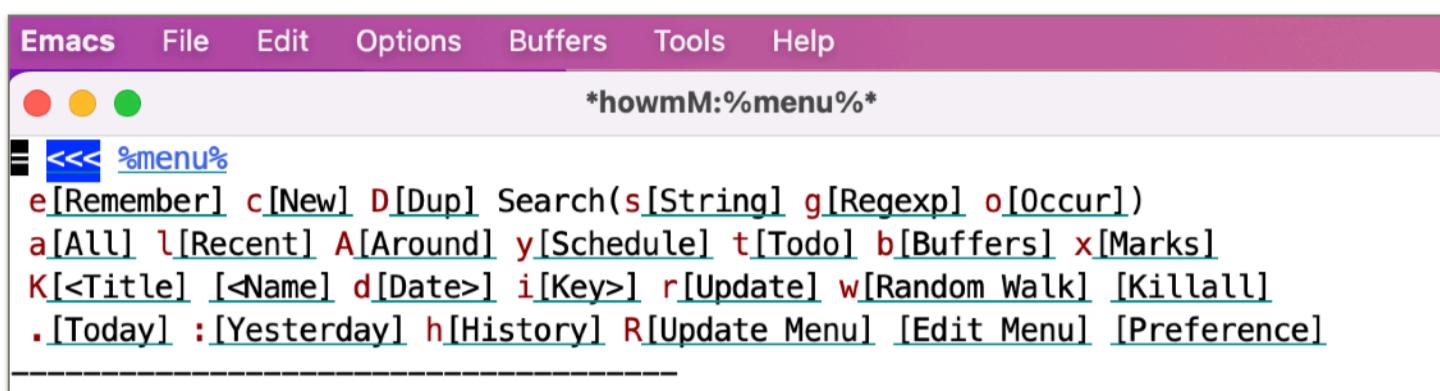
⁹ Конечно, вам не обязательно каждый раз заходить в меню для взаимодействия с `howm`. Это, скорее, справочная система. Любые действия можно выполнить с помощью горячих клавиш, как только вы запустите `Emacs`.

¹⁰ На MacBook это клавиша `Delete`.

Развернутый вид на начальную страницу howm выглядел бы так:



Теперь давайте обстоятельно разберем, какие действия можно совершить из основного меню.



Будем двигаться побуквенно, раз уж каждая команда привязана к определенной клавише.

e – создание быстрой записи-памятки (открывается документ без шаблона).

c – создание новой заметки.

D – дублирование файла. Полезно, если вы используете шаблоны. По умолчанию дублируется последняя запись из списка открытых буферов.

s – поиск по ключевым словам, плюс обычный поиск.

g – обычный поиск, плюс поиск с помощью регулярных выражений.

o – поиск в последнем открытом буфере. То есть, вместо поиска по всей картотеке, поиск будет проведен в одной заметке.

a – вызов списка всех заметок (по умолчанию в порядке правок).

l – вызов списка заметок, с которыми вы недавно работали.

A – вызов заметок в порядке их создания.

y – вызов расписания, если у вас что-то запланировано.

t – вызов списка дел.

b – вызов списка открытых буферов.

x – открывает последний файл, в котором вы разместили метки, все метки будут выведены списком в верхнем окне.

K – копирование заголовка последней заметки из списка открытых буферов.

[<Name>] – копирование нумерического имени последней заметки из списка открытых буферов (копируется полный путь к заметке).

d – вставляет дату в заметку (в последнюю запись из списка открытых буферов). Функция полезна при планировании.

i – вставка ключевого слова (в последнюю запись из списка открытых буферов).

r – обновление заметок, в которых есть «ссылки-сюда». Если в каком-то тексте появились ключевые слова, они будут активированы¹¹. Кроме того эта команда перебрасывает вас из меню в последний открытый буфер. Если вы заглянули в меню из заметки, можно быстро к ней вернуться.

w – просмотр заметок путем случайного перехода по ссылкам. Цикл бесконечен. Остановить перелистывание можно нажав **C-g**

[Kill all] – убирает из списка открытых буферов все заметки `howm`, кроме несохраненных.

. – открывает список заметок, которые были созданы сегодня.

: – открывает список заметок, которые были созданы вчера.

h – список с историей поисковых запросов.

R – обновление меню. На самом деле любой вызов меню обновляет его. Эта команда полезна, если меню открыто у вас отдельном фрейме или вкладке.

[Edit menu] – переход к настройками меню (можно удалить или добавить элементы).

[Preference] – переход к настройкам `howm`

Как я уже сказал, все команды, расположенные в меню, доступны и благодаря горячим клавишам. Их можно вызвать даже в процессе работы с другими файлами за пределами `howm`¹². Просто нужно будет предварить символ, указанный в меню, следующим клавиатурным сочетанием: **C-c ,**

Например, функция «блуждания» вызывается командой **C-c , w**

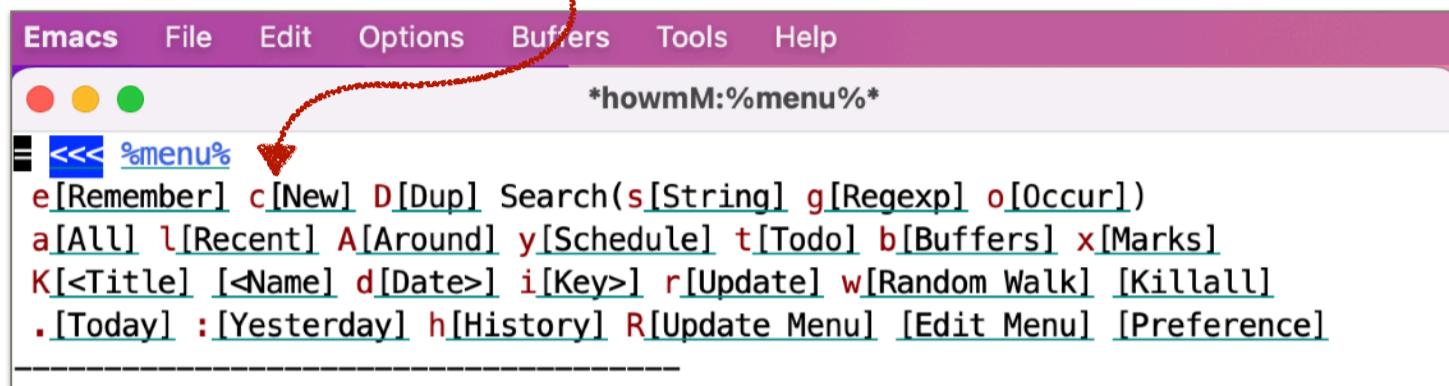
¹¹ Подробнее об этом в главе «Ключевые слова в `howm`».

¹² `Howm` отзывается на команды из основных режимов. Если вы запустили какой-то побочный режим (`minor-mode`), `Howm` нужно будет активировать: `M-x howm-mode`
В общем, если команда не срабатывает, проверьте, активирован ли `howm`.

СОЗДАНИЕ ЗАМЕТОК

Давайте создадим первую заметку.

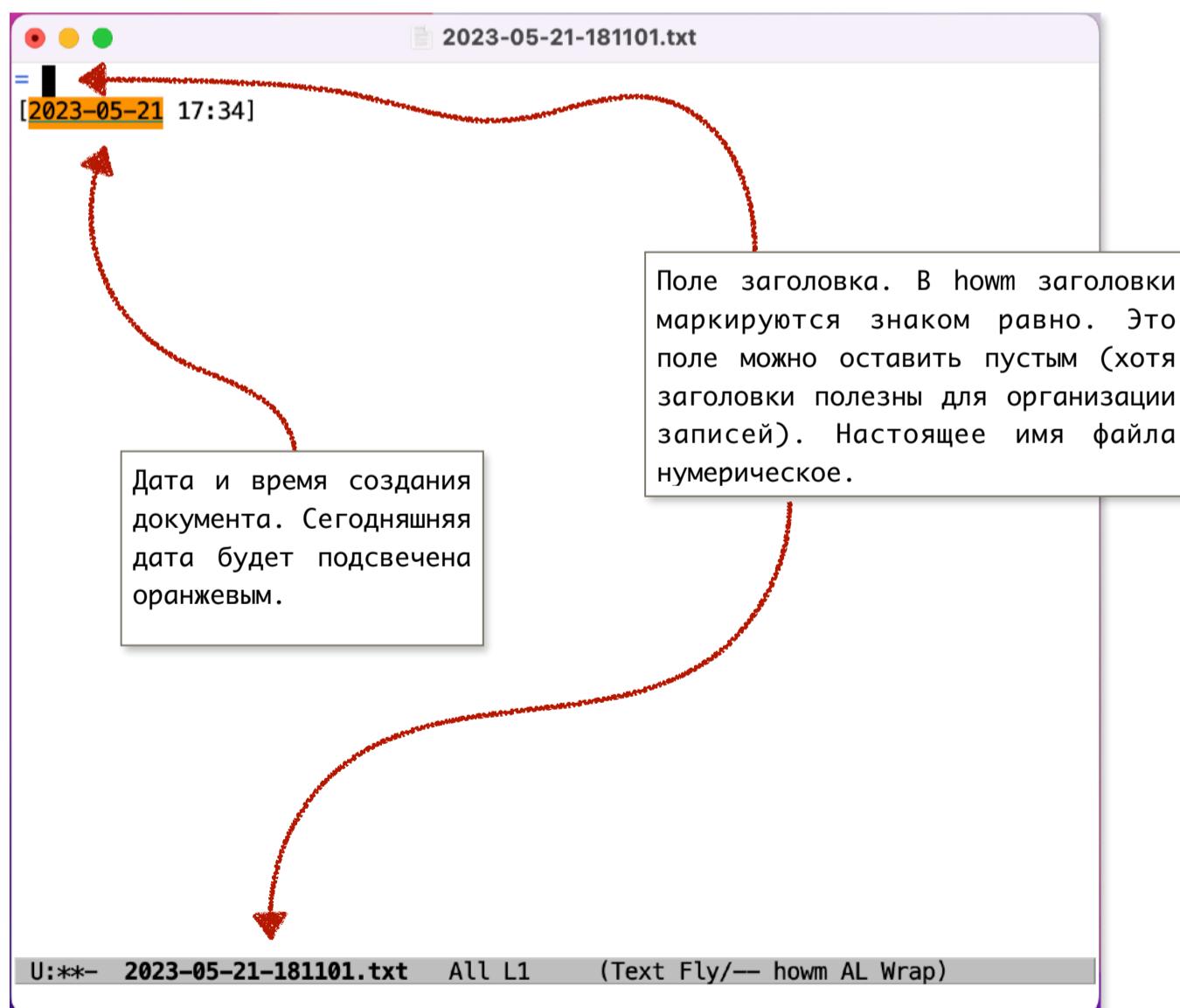
Если вы в основном меню `howm`, достаточно переместить курсор на соответствующую кнопку `[New]` или просто нажать клавишу `c`. Она, понятное дело, соответствует начальной букве слова `create`.



Вы можете выполнить это же действие, минуя основное меню, командой `C-c , c`

Сразу после запуска Emacs наберите это сочетание клавиш, и `howm` откроет для вас окно новой заметки.

Что мы в тут имеем?



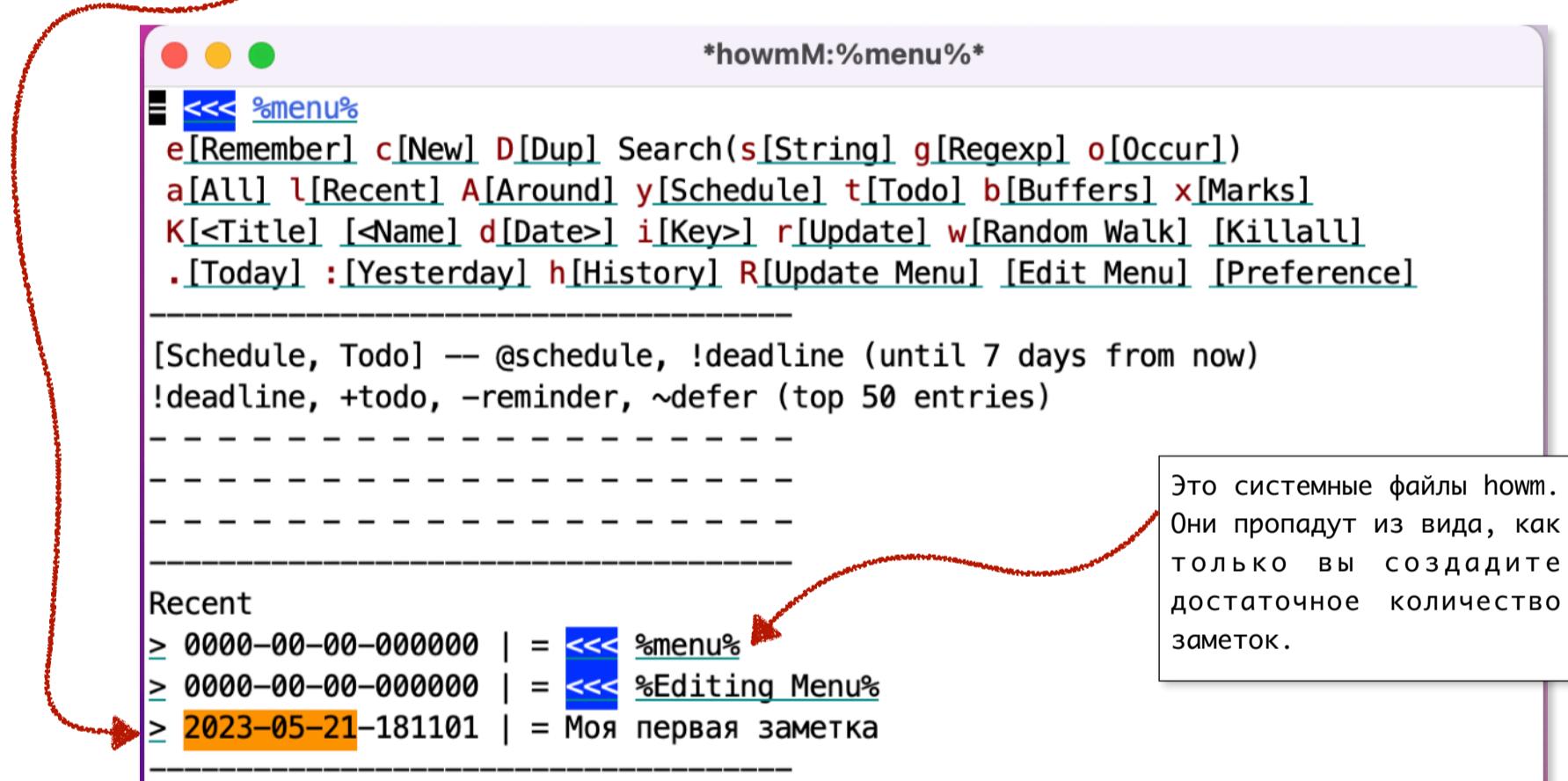
Уникальным идентификатором заметки является нумерическое имя. Поэтому, заголовок может быть идентичным у разных заметок (хотя, это плохая практика) или вовсе отсутствовать.

Чтобы создать заметку спуститесь курсором ниже даты и начните печатать.

По умолчанию `howm` сохраняет записи в `txt` формате, поэтому никакие косметические манипуляции с текстом (выделение жирным, подчеркивание и т.п.) невозможны¹³.

Создав первую заметку, сохраните ее стандартной для Emacs командой - `C-x C-s`

Если вы перейдете в основное меню, то увидите, что она появилась в списке `Recent`.



Для того, чтобы вернуться к заметке из основного меню можно переместить курсор на соответствующую строку и нажать RET.

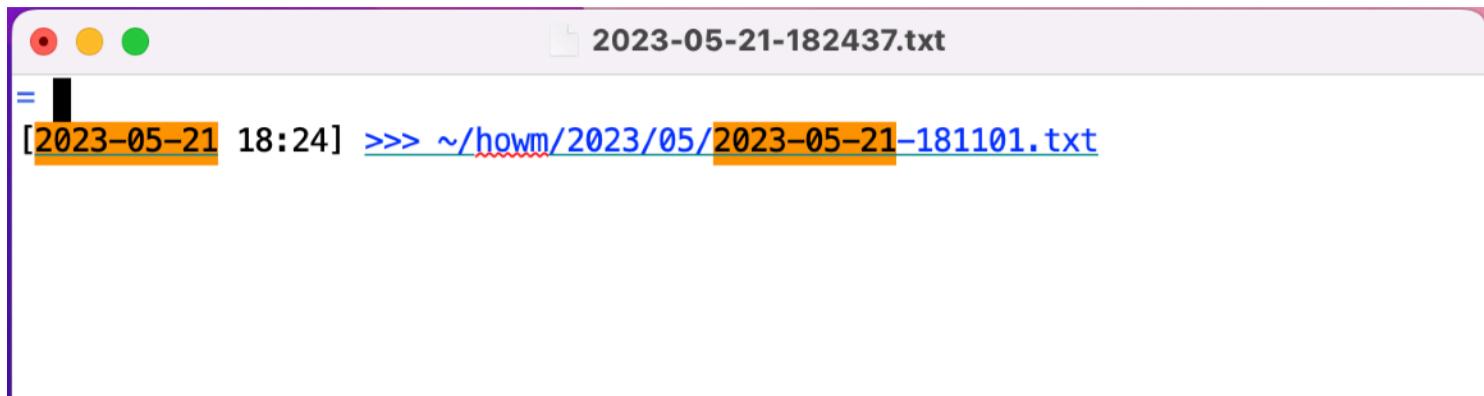
В процессе написания первой заметки вас могут посетить идеи, развивающие тему.

Есть искушение писать их тут же, превращая текст в бесконечный свиток. Лучше, однако, оставлять заметки модульными.

Создайте еще одну заметку все той же командой `C-c , c`

¹³ Вы можете поменять расширение файла (см. раздел «Дополнительные настройки», стр. 79). Плюс, если вы разместите картотеке документ `org`, его специфичное форматирование будет по-прежнему функционировать. `Howm` дружит с любыми побочными режимами (`minor modes`).

Обратите внимание, что шаблон новой заметки чуть отличается. Рядом с датой появилась ссылка на предыдущую запись¹⁴.

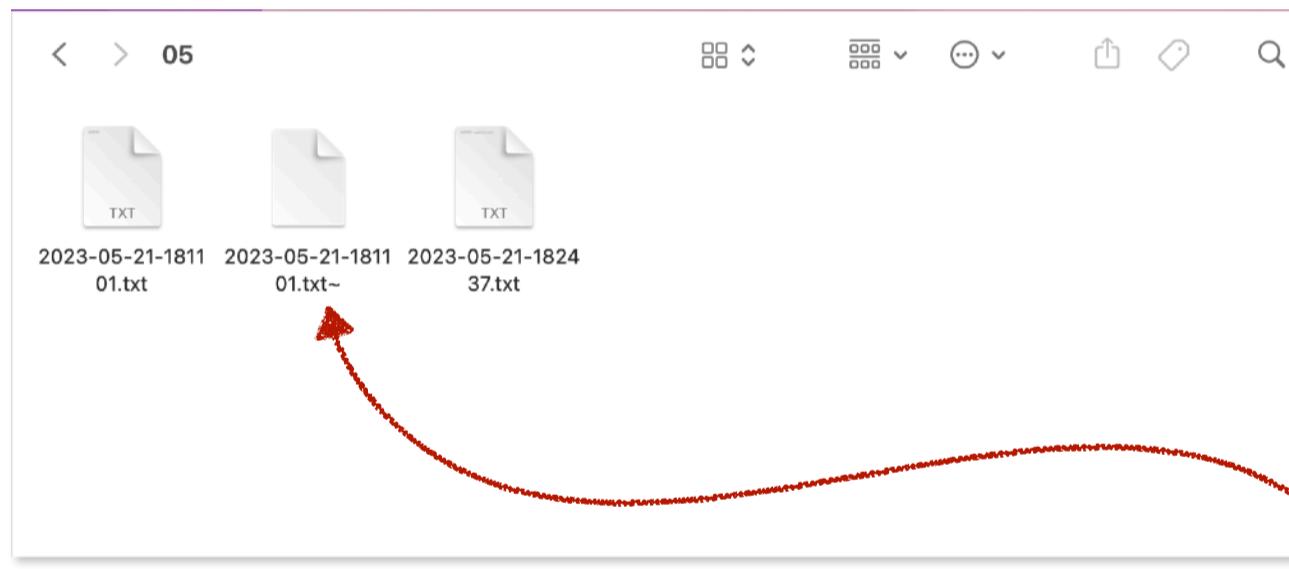


Всякий раз, когда вы, находясь в одной заметке, создаете другую, howm их автоматически линкует¹⁵. Подручное средство для ассоциативного мышления.

Если вам эта ссылка не нужна, используйте команду отмены - С-/

Если вы и вовсе передумали создавать заметку, нажмите на отмену еще раз.

К слову, о сохраненных заметках. Они складируются в папке howm в домашней директории. Howm создает подпапку для каждого года и под-подпапку для каждого месяца¹⁶.



Рядом с каждым файлом вы можете заметить его клон со странным хвостиком. Не пугайтесь, - это нормальная практика Emacs - создавать резервную копию. Очень полезно, если что-то пошло не так. Карман эти файлы не тянут, глаз не мозолят. Мы в эту папку заглянули в экскурсионных целях.

В какой-то момент заметок у вас станет много. Как открыть нужную? Как сделать обзор записей?

В howm эти задачи решаются с помощью списков.

¹⁴ О том, как работают ссылки в howm разговор пойдет чуть дальше.

¹⁵ Ссылка появляется только во второй заметке.

¹⁶ Путь к файлу: ~/howm/год/месяц/имя файла. Все это закулисье программы. В интерфейсе папки не видны.

РАБОТА СО СПИСКАМИ

Команды вызова недавних заметок, вызова всех заметок, поиска и даже перехода по ссылкам переводят howm в двухоконный режим. Большая часть манипуляций с файлами осуществляется благодаря этому интерфейсу. Разберем его «анатомию».

The screenshot shows the 'howmS' application window. The top bar has three buttons (red, yellow, green) and the title '*howmS*'. The main area contains a list of file names on the left and their contents on the right. A red arrow points from the top-left of the list area to a callout box. Another red arrow points from the bottom-left of the list area to a callout box. A third red arrow points from the bottom-right of the preview area to a callout box. The preview area at the bottom shows the content of the file '2023-04-14-182224.txt'.

Слева в верхнем окне находится колонка с нумерическими именами заметок. Они подсвечены голубым. Если какие-то заметки созданы сегодня, они будут подсвечены оранжевым.

Справа при вызове списка отображаются заголовки заметок, если они созданы. Если вы решите не заполнять заголовок, в списке вместо него будет отображена первая строка текста.

В нижнем окне находится предпросмотр заметки. Отображается заметка, на которую наведен курсор в списке. При перемещении курсора, отображение в нижнем окне будет динамически меняться.

U:%%- *howmS*

2023-04-14-190528.txt | = Философия как психотерапия
2023-04-14-182646.txt | = Соблазн самосовершенствования
2023-04-14-184903.txt | = Дневник и проблема самокопания
2023-04-14-183747.txt | = Быть человеком
2023-04-14-183154.txt | = Несовершенство мира
2023-04-14-182224.txt | = Нет хуже человека
2023-04-14-182039.txt | = Не мучь себя
2023-04-14-160713.txt | = Десять заповедей изучения языков
2023-04-14-133721.txt | = Другой язык, другая личность
2022-10-06-111115.txt | = Накопление знаний
2023-04-13-144303.txt | = Реалистичные сроки освоения языка
2023-04-13-142613.txt | = Минимум в изучении языков
2023-04-13-143504.txt | = Ломб об изучении слов
2023-04-13-143407.txt | = Словарь как помеха
2022-10-18-143554.txt | = Компьютерные языки
2023-02-13-120000.txt | = Изучение языков и научная деятельность
2023-04-13-142326.txt | = Чтение как главный инструмент в изучении языков
U:%%- *howmS*

25% L192 (HowmS)

Нет хуже человека

[2023-04-14 18:22] >> ~/howm/2023/04/2023-04-14-182039.txt

«Кто зол для себя, для кого будет добр? И не будет он иметь радости от имения своего. Нет хуже человека, который недоброжелателен к самому себе...» (Сир. 14:5–6).

U:**- 2023-04-14-182224.txt All L1 (Text Fly/-- howm AL Wrap)

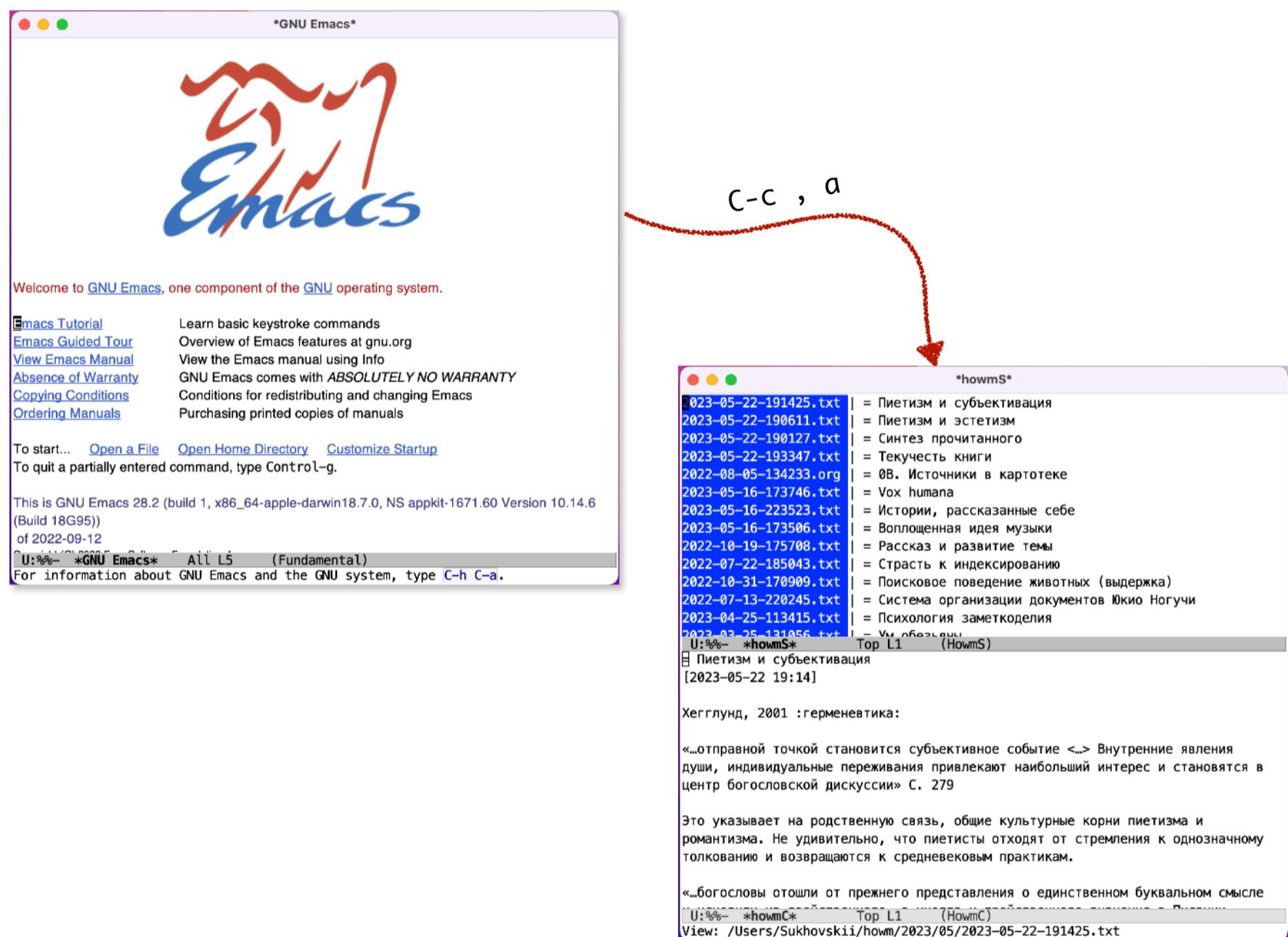
Поскольку списки – это ключевой элемент в работе с `howm`, стоит подробно рассмотреть функции и возможности, скрытые здесь. Какой бы список пред вами ни был, принципы работы с ним будут одинаковыми.

Из основного меню `howm` списки заметок можно вызвать командами:

`a[All]` `l[Recent]` `A[Around]`

Однако вам не обязательно находиться в основном меню, чтобы вызвать список. Намного удобнее использовать для этого специальные команды.

Давайте откроем список всех заметок – `C-c , a`



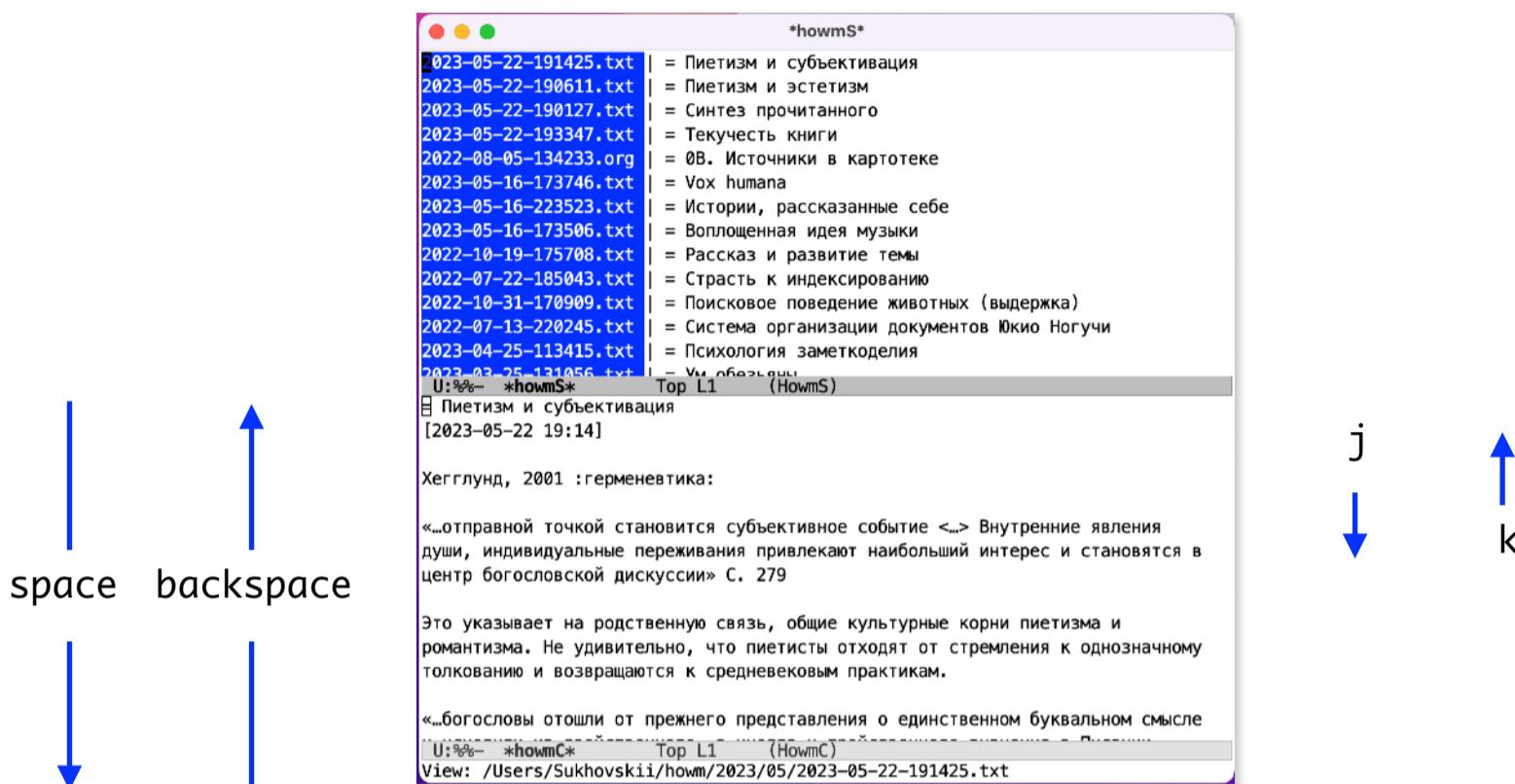
Поговорим сначала о командах навигации.

Перемещение курсора по списку осуществляется клавишами `p` и `r`, вниз и вверх соответственно¹⁷. В нижнем отсеке по мере вашего движения перелистываются

¹⁷ Стандартные перемещения Emacs `C-p` и `C-r` тоже работают. Лучше использовать их для тренировки моторной памяти.

заметки. Также можно использовать клавишу табуляции для перемещения к следующей строке списка, а табуляцию с альт для перемещения наверх.

Зачастую все содержимое заметки не влезает в окно с предпросмотром. Вы можете скролить текст вниз, нажимая пробел, и вверх нажимая backspace¹⁸. Для медленного построчного прокручивания текста используйте клавиши j и k.



Если вы хотите перейти в нижнее окно с заметкой, нажмите на RET. Курсор при этом переместится в заметку. Теперь вы можете внести необходимые дополнения или правки.

Заметьте, что верхнее окно со списком останется открытым. Это удобно для разного рода быстрых правок и других манипуляций. Вы можете вернуться к списку с помощью команды С-x o. Это стандартный для Emacs способ переключения между окнами.

Что если вы хотите открыть заметку и закрыть список? Переместите текстовый курсор на нужную строку в списке и нажмите С-и RET¹⁹.

Что если вам нужно прямо противоположное? Вы хотите увидеть длинный список заметок без окна предпросмотра. Его можно закрыть, просто нажав 1. Тогда перед вами будет только список²⁰. Чтобы вернуть предпросмотр, нажмите 2. Вы также можете переключаться туда-сюда, нажимая v.

¹⁸ На Macbook это клавиша delete. Не бойтесь в данном случае, ничего не будет удалено.

¹⁹ Можно также перейти в заметку и нажать С-x 1. Это стандартная для Emacs команда, которая закрывает все окна, кроме текущего.

²⁰ Для корректной работы этой функции нужно прописать в .emacs следующую строчку:
(setq howm-view-keep-one-window t)

Подробнее об этом в разделе «Дополнительные настройки», стр. 80

Теперь давайте обратимся непосредственно к работе со списками.

Я уже сказал, что команда `C-c`, а вызывает список всех файлов, размещенных в `howm`. По умолчанию они рассортированы в порядке последнего редактирования. То есть сверху будут находиться файлы, с которыми вы недавно работали²¹.

Режим общего списка позволяет вам осуществить несколько полезных операций.

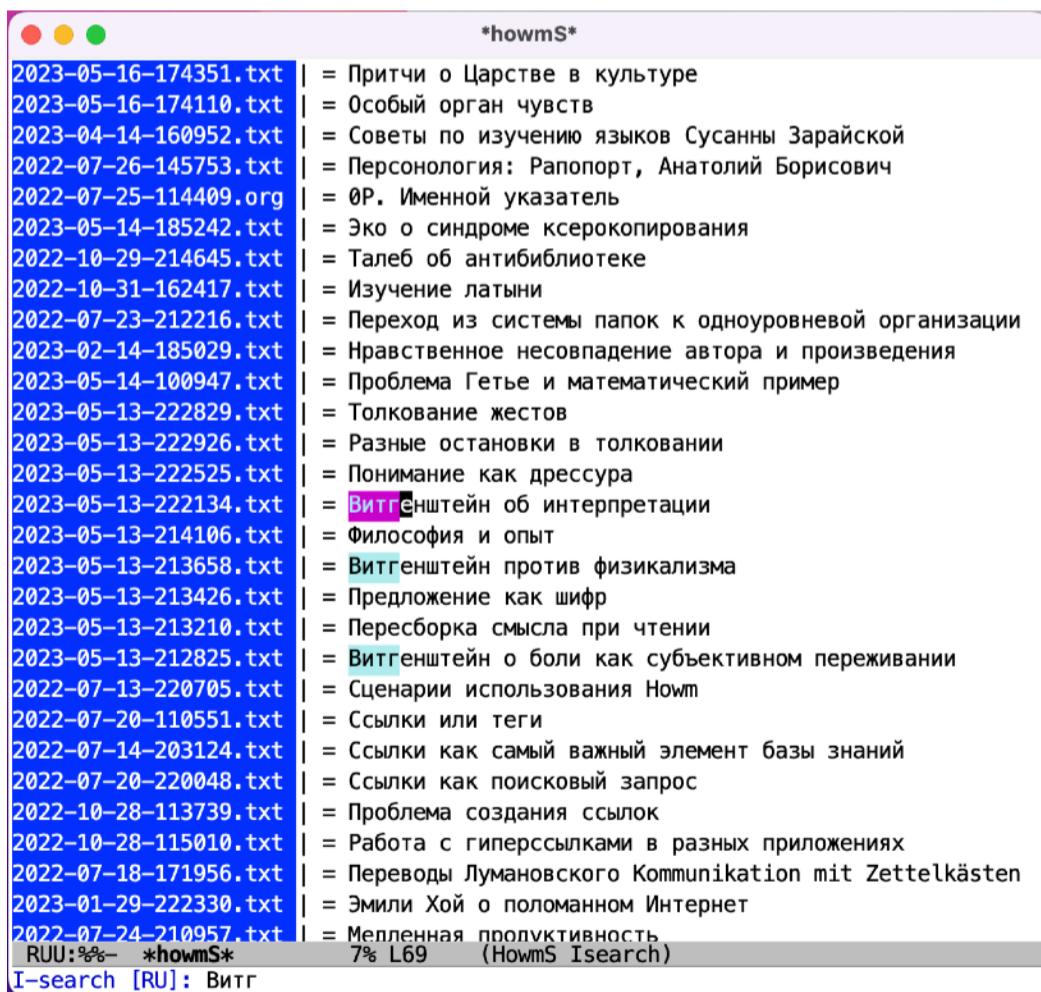
Подсчет заметок

Наберите команду подсчета строк - `C-x l`, чтобы узнать, сколько заметок в вашей картотеке. Цифра появится в нижнем минибуфере²².

Быстрый переход к заметкам

Общий список может быть полезен для быстрого перехода заметке, заголовок которой вы примерно помните. Тут удобнее переключиться в режим «только список».

Нажмите `C-s` - стандартный поиск по тексту в Emacs - и начните вводить название заметки. Можно даже не с первого слова. Для перехода от между результатами поиска используйте ту же команду `C-s`.



²¹ Шанс, что вам потребуются именно они очень высок. Именно на этом построена система самоорганизации Юкио Ногучи.

²² Поскольку в общем списке есть две строки из системного файла `howm`, их нужно вычесть из получившегося числа.

Кроме общего списка, вы можете запросить у howm список недавних заметок – **l**[Recent] в основном меню.

Для вызова используйте клавиатурное сочетание – **C-c , l**

По этой команде howm покажет заметки, созданные или отредактированные за последние семь дней²³.

Список позволяет быстро переключаться между файлами, с которыми вы работаете в данный момент.

Нужен список длиннее?

Вот пример, как можно запросить у howm сводку за 20 дней – **C-u 20 C-c , l**

Используя комбинации этой команды, вы можете посмотреть файлы с которыми работали в этом месяце или квартале. Это полезно для поведения итогов, самоанализа.

Команда **A**[Around] в основном меню вызывает список заметок, рассортированных по дате их создания.

Находясь в открытой заметке, вы можете запросить тот же самый список командой **C-c , A**. Отличие в том, что список получится контекстным, центрированным на текущей заметке.

Список с курсором на текущей заметке. Она посередине. Ее окружают заметки, созданные до и после. Порой это полезно, чтобы восстановить ход мысли.

howmS

2023-05-20-235011.txt	= Невозможность со-бытия Аврааму
2023-05-20-234757.txt	= Невозможность со-бытия Иову
2023-05-20-234124.txt	= Сперджен и аллегорическое толкование
2023-05-20-192202.txt	= Крушение церковных надежд
2023-05-20-190658.txt	= Птица в клетке
2023-05-20-190453.txt	= Толкин о проповедях
2023-05-20-143409.txt	= Довлатов о юморе как защите
2023-05-18-222143.txt	= Поварнин о картотечном коллекционерстве
2023-05-18-172538.txt	= Цинодку
2023-05-17-234237.txt	= Теопоэтика в Новом Завете
2023-05-17-110014.txt	= Повторы у Довлатова
2023-05-16-231504.txt	= Адорно об антропоморфизме Просвещения
2023-05-16-230503.txt	= Одна история везде и всегда
2023-05-16-230511.txt	= Есть только одна история

U:%%- *howmS* 2% L18 (HowmS)

Поварнин о картотечном коллекционерстве
[2023-05-18 22:21] >> ~/howm/2023/05/2023-05-18-172538.txt

Поварнин, 2015 :карточка:

«Не мешает здесь подчеркнуть, что выписки на карточках и в тетрадях – только вспомогательное средство при чтении, а не цель сама по себе. Иной склонен обратить это в какое-то коллекционерство, в своего рода "спорт" – гордится своей "карточкой" выпуск и выдержек». С. 691 Сочинения

U:%%- *howmC* All L1 (HowmC)
View: /Users/Sukhovskii/howm/2023/05/2023-05-18-222143.txt

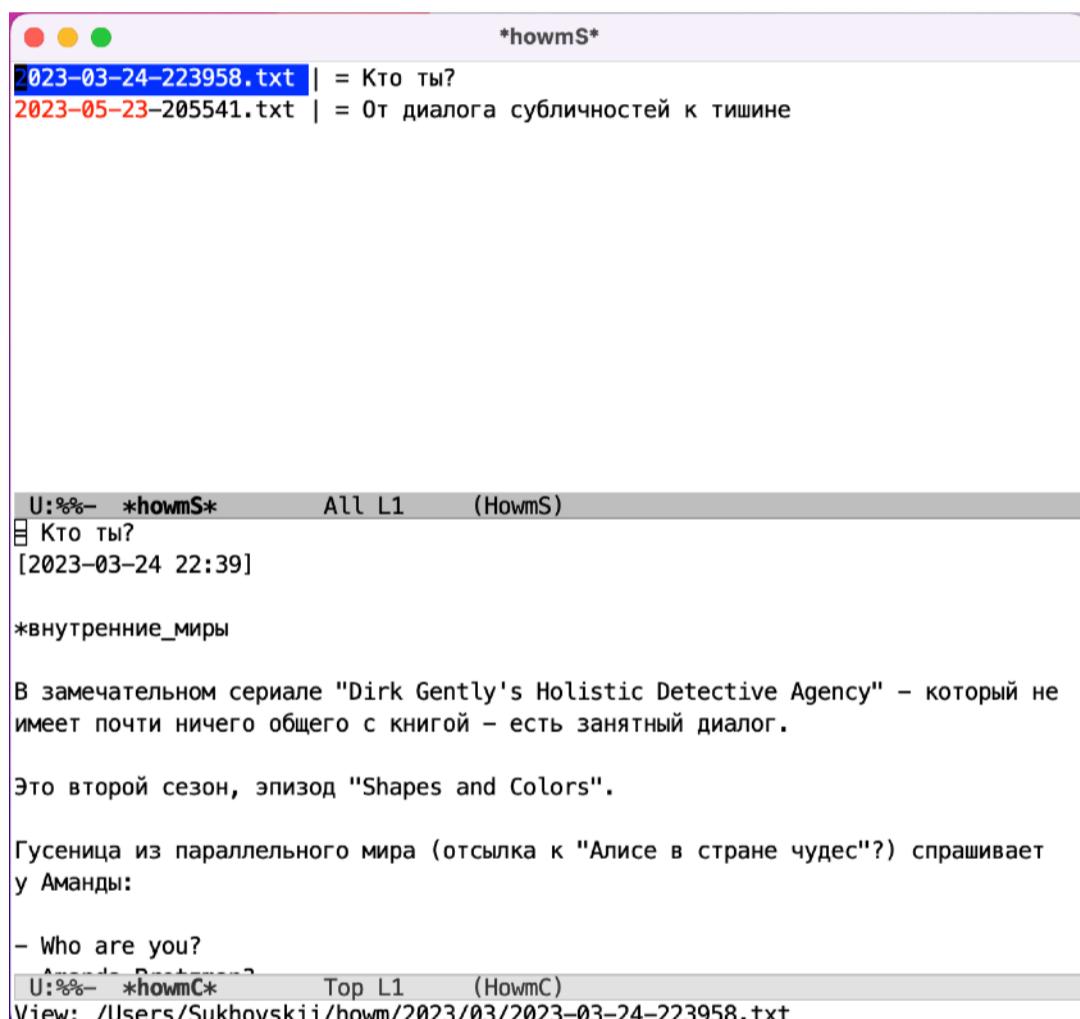
²³ Заметьте, именно за 7 дней. Если вы сделаете запрос в четверг, то в списке будут все измененные/созданные заметки с прошлого по этот четверг.

Что если на склоне дня вы захотели посмотреть все заметки, которые сегодня были внесены в базу? Конечно, они и так подсвечены оранжевым.

Однако, если вам хочется отдельный список, используйте команду `C-c ,`.

Заметьте, что на экран будет выведен лишь список созданных заметок (отредактированные заметки в данном случае не учитываются).

Если вы используете нумерические ссылки²⁴, в списке окажутся и старые заметки, которые вы залиновали с сегодняшними. Причина в том, что `howm` ищет заметки по дате и в названии файла, и в самом тексте. Старые заметки будут помечены синим цветом, сегодняшние - на белом фоне²⁵.



На самом деле, если вы разместите курсор на дате в квадратных скобках в заголовке заметки и нажмете RET, появится диалог минибуфера.

```
U:--- 2023-04-10-140557.txt All L2 (Text Fly/-- howm AL Wrap)
[Mon] RET(list), +num(shift), yymmdd(set), ~yymmdd(repeat), .(today): █
```

Повторное нажатие RET вызовет список заметок, ассоциированных с этой датой²⁶.

²⁴ О них речь пойдет дальше.

²⁵ Это же относится к заметкам с планами (о них позже). Если вы вызовете список заметок на сегодня, то заметите, что в него включены старые записи, которые содержат задачи с нынешней датой.

²⁶ Мы рассмотрим остальные опции на стр. 73

Хотите посмотреть список заметок, созданных вчера? Наберите команду С-с , :

Возможно вам интересны заметки, которые вы создали какое-то количество дней назад (возьмем для примера 50 дней).

Тогда на выручку приходит команда С-и 50 С-с , .

Как только вы вызвали список заметок, связанных с определенной датой, вы можете перемещаться по ним с помощью горячих клавиш.

(, { , [- обратно на день, месяц и год.

) , } ,] - вперед на день, месяц и год.

Таким образом вы можете легко проверить, о чем вы писали ровно год или два назад.

СОРТИРОВКА ЗАМЕТОК В СПИСКАХ

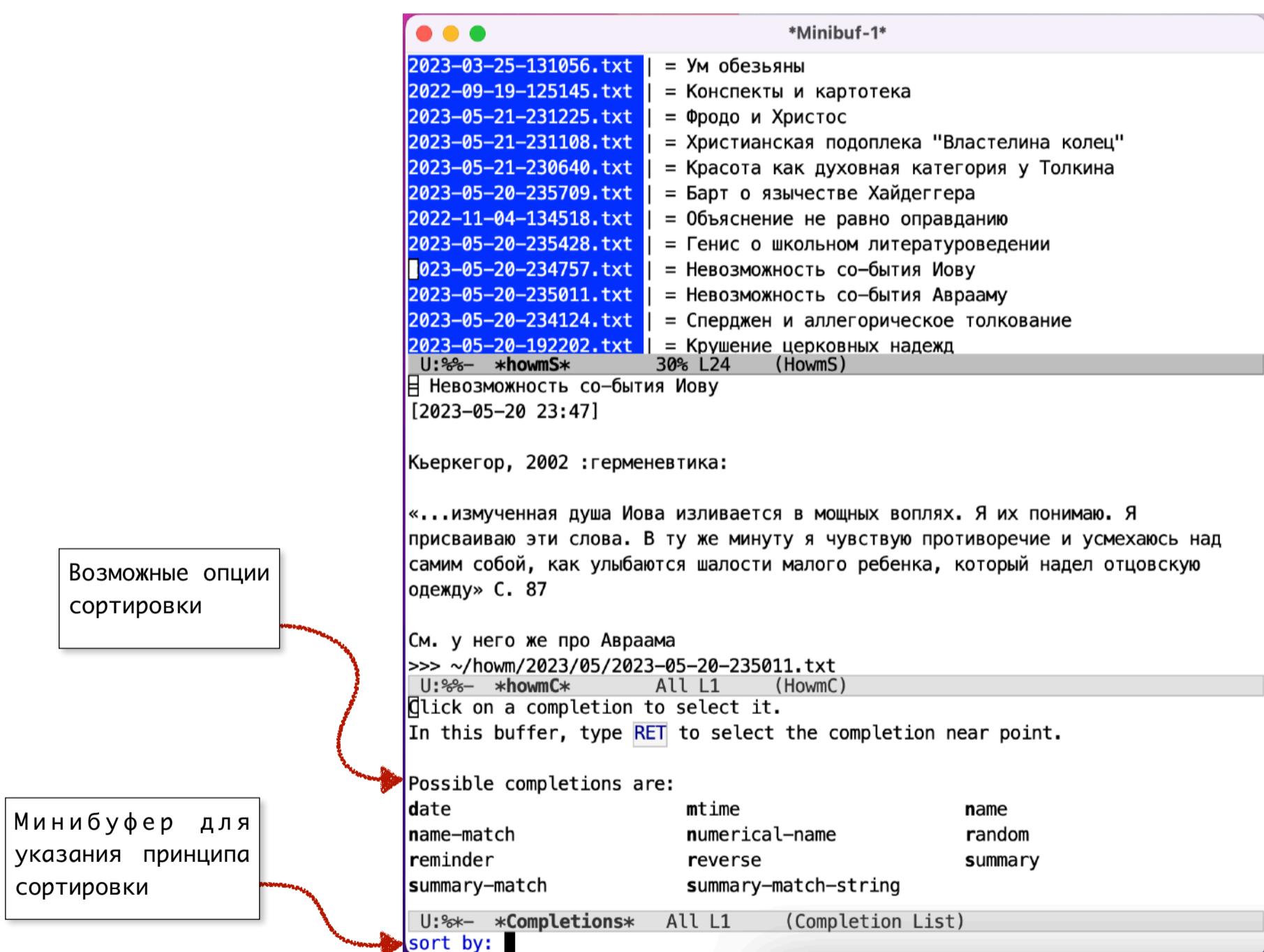
Запросив у howm список заметок командой `C-c , a` или `C-c , l`, вы получаете его рассортированным в порядке свежести правок. То есть наверху будут заметки, с которыми вы недавно работали.

Что если вы хотите какие-то иные формы выстраивания записей?

Нажмите `S`

Howm спросит вас, по какому принципу вы хотите расположить заметки.

Дальше нажмите `Tab` для того, чтобы посмотреть варианты²⁷.



Возможно не все опции вам пригодятся, но некоторые, безусловно, способны повысить эффективность навигации по заметкам.

²⁷ Эти команды бодрее набирать, если у вас дополнительно установлен пакет Vertico.

date

Функция позволяет сформировать список заметок по дате создания. Сверху у вас будут самые свежие записи.

Чтобы увидеть список в обратном порядке (от старых к новым), нажмите **R**. Чтобы вернуться обратно нажмите **R** еще раз. Команда **R** переворачивает любой список. То что было внизу, окажется наверху.

Если вам сразу нужен обратный порядок правок, дат создания или алфавита, наберите команду **C-u S**, а потом выберите нужную опцию.

reverse

Выполняет ту же функцию переворачивания списка вверх тормашками. Легче и быстрее пользоваться клавишей **R**.

summary

Справа от нумерического имени файла в списке находится заголовок (если вы его создали). Команда **summary** нужна, чтобы рассортировать заметки в алфавитном порядке, по заголовкам.

В заголовки, как вы догадываетесь, можно запихать дополнительный слой метаданных. Используя определенные префиксы перед названием заметки, вы получаете подобие Лумановского *folgezettel*. Отмечу, однако, что усложнение системы редко идет на пользу.

mtime

Если вы после всех этих манипуляций захотите вернуться к списку по времени модификации, наберите команду **mtime**. Хотя можно просто перезапустить процесс, вызвав нужный список еще раз.

random

Надоел порядок? Растирайте записи, используя команду **random**. Возможно, в этом случае вам на глаза попадется подзабытая, но важная заметка²⁸.

name-match

Благодаря этой команде, вы можете расположить заметки в удобной для вас очередности²⁹. **Howt** попросит вас указать имя файла, который необходимо

²⁸ Напомню, что на начальной странице **howt** уже есть список случайных заметок.

²⁹ Функция произвольного распределения чрезвычайно полезна при редактуре заметок и сшивании их в единый текст. Об этом мы поговорим чуть дальше.

переместить наверх списка. Речь в данном случае о нумерическом имени. Можно указать лишь его часть.

summary-match

Такую же операцию перераспределения можно осуществить и по заголовкам. Заметьте, что можно указать часть заголовка - например, какое-нибудь слово - и Howm переместит наверх все заметки, где это слово в заголовках упоминается.

summary-match-string

Для тех, кто разбирается в регулярных выражениях, есть возможность более хитрых фильтров³⁰.

name

Команда name позволяет рассортировать файлы с нецифровым именем. Где это может пригодится?

Заметки, созданные в howm, автоматически получают нумерическое имя, основанное на дате. Однако, что если вы хотите забросить в чрево howm другие текстовые файлы? Howm их без труда переварит. Команда name выстраивает такие заметки в алфавитном порядке. Они окажутся в конце списка, так как по очередности сначала идут цифры, а потом буквы³¹.

reminder

Команда позволяет вывести вверх списка задачи, если таковые имеются. Мы к этой функции вернемся, когда будем говорить об организации дел в howm. Здесь довольно гибкая система планирования.

numerical-name

Команда, которая позволяет сортировать корреспонденцию, если вам вдруг захочется совместить базу знаний с почтовым клиентом³².

³⁰ Использование Regex - это отдельная тема, выходящая за пределы данного краткого пособия.

³¹ Вам даже не обязательно забрасывать инородные файлы в howm. Вы можете просто указать путь к ним. См. раздел «Дополнительные настройки», стр. 80

³² Если вы используете почтовый клиент, такой как Wanderlust, можно настроить howm на поиск файлов с соответствующей директории. См. раздел «Дополнительные настройки», стр. 80.

РАБОТА СО ССЫЛКАМИ

Картотека это не просто собрание заметок, но система связей между ними. В howm они реализуются с помощью гиперссылок. Однако механизм работы у них не совсем обычный.

Вкратце, ссылка в howm это триггер поискового запроса. Наведя курсор на ссылку и нажав RET, вы инициируете поиск. Экран при этом разделяется на две части. В верхней результаты поисковой выдачи, в нижней просмотр заметки, на которую наведен курсор. То есть вас не сразу перебрасывает из одной заметки в другую. На этом промежуточном этапе вы можете взглянуть на место заметки в сетевой структуре.

Давайте добавим наглядности и разберем конкретные примеры.

Синтаксис ссылок в howm предельно прост.

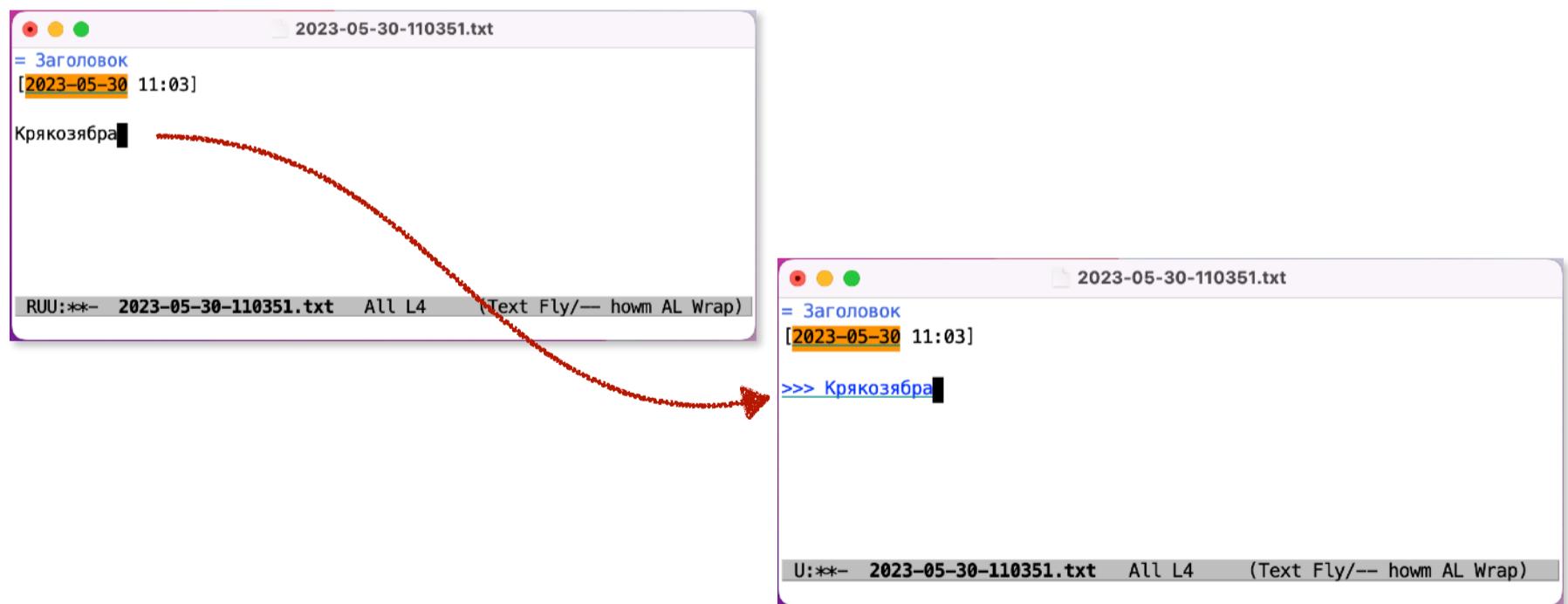
>>> - ёлочка, повернутая вправо.

<<< - ёлочка, повернутая влево³³.

Они отличаются по принципу действия, поэтому стоит рассмотреть их отдельно.

Эта глава будет посвящена ёлочке-вправо. А следующая – ёлочке-влево.

Добавив значок ёлочки перед словом или фразой, вы создаете ссылку³⁴.



³³ Есть также возможность использовать традиционные для Markdown квадратные скобки. Об этом чуть дальше.

³⁴ Лучше добавить после ёлочки знак пробела для лучшей читаемости.

Все, что расположено на строке правее значка >>> howm будет интерпретироваться как ссылку/поисковый триггер. Поэтому для набора текста после ссылки нужно перейти на новую строку.

Вы можете создать ссылку на конкретный заголовок или на нумерическое имя файла. Результат будет несколько отличаться. Рассмотрим последовательно оба варианта.

The screenshot shows three windows of the Howm application:

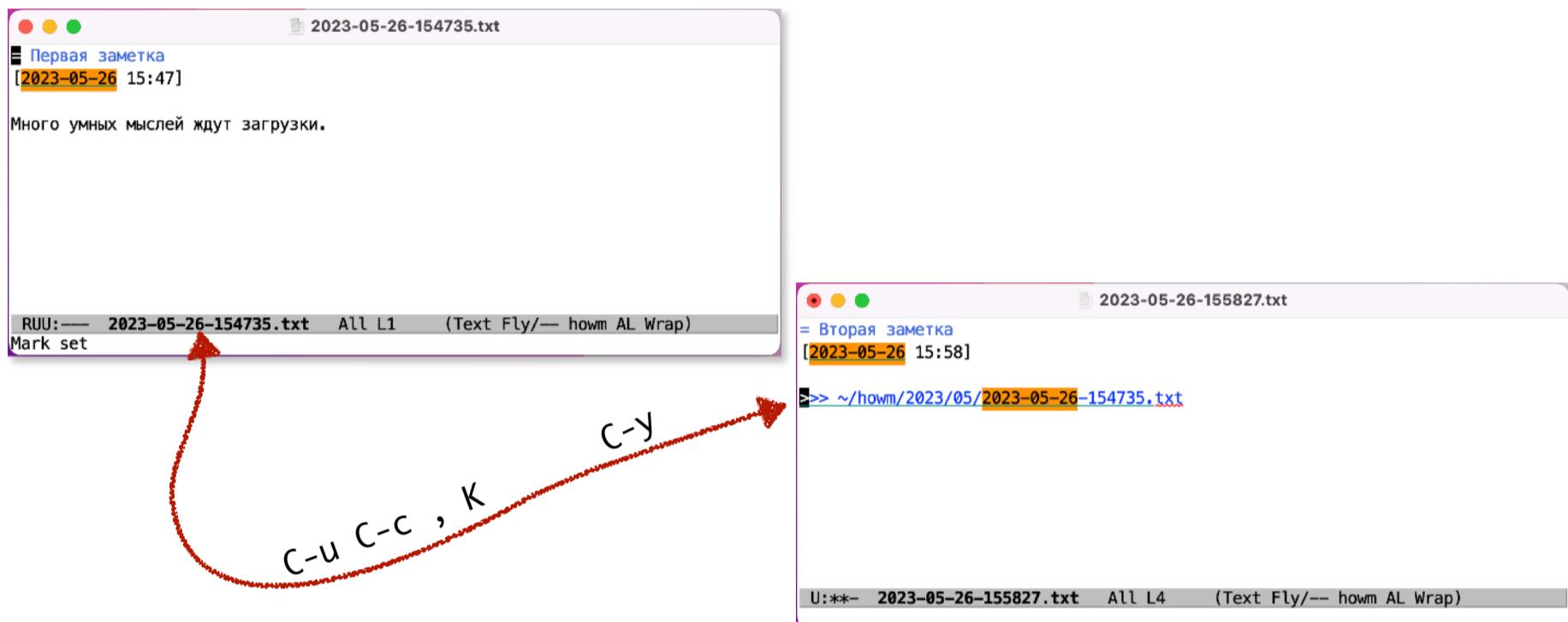
- Top Left Window:** Title bar: 2023-05-26-154735.txt. Content: '= Первая заметка [2023-05-26 15:47]'. Below it: 'Много умных мыслей ждут загрузки.' A red arrow points from the text '= Первая заметка' to the right, indicating where to type the link name.
- Top Right Window:** Title bar: 2023-05-26-155827.txt. Content: '= Вторая заметка [2023-05-26 15:58]'. Below it: '>>> Первая заметка'. A red arrow points from the text '= Вторая заметка' down to the link text.
- Bottom Window:** Title bar: *howmS*. Content: 'RUU:--- 2023-05-26-155827.txt All L4 (Text Fl)' at the top. Below it: '2023-05-26-155827.txt | >>> Первая заметка' and '2023-05-26-154735.txt | = Первая заметка'. At the bottom: 'U:%%- *howmS* All L1 (HowmS)' and '>>> Первая заметка'.

Annotations:

- Left Annotation:** 'Найдясь в первой заметке, нажмите С-с , К' (While in the first note, press S-s, K).
- Middle Annotation:** 'Перейдите во вторую заметку. Наберите >>> и нажмите С-у' (Go to the second note. Type >>> and press S-u).
- Right Annotation:** 'Добавьте во вторую заметку ссылку на первую. Для этого достаточно записать название заголовка после ёлочки-вправо.' (Add a link to the first note in the second note. To do this, just write the title name after the right-angle bracket).
'Можно вписать от руки, но легче скопировать и вставить.' (It can be handwritten, but it's easier to copy and paste).
- Bottom Annotation:** 'Поместите курсор на ссылку и нажмите RET.' (Place the cursor on the link and press RET).
'Вас перебросит в двухоконный режим (список и окно предпросмотра).' (It will switch to a two-window mode (list and preview window)).
'Вы увидите заметку со ссылкой >>>' (You will see a note with the link >>>).
'И заметку с определенным заголовком (она со знаком равно).' (And a note with a specific title (it has an equals sign)).
'Фактически howm вывел на экран список файлов, где встречается словосочетание «Первая заметка».' (In fact, howm displayed on the screen a list of files where the phrase 'Первая заметка' appears).
'В выдаче будут все заметки, которые где-либо содержат подобное словосочетание. Это могут быть как ссылки, так и просто упоминания.' (In the output, all notes will be shown that contain this phrase somewhere. It can be links or just mentions).

Другой способ создания ссылки – использование нумерического имени файла. Такая ссылка надежнее. В предыдущем случае, поменяв заголовок, вы наплодите битых ссылок.

По команде `C-u C-c , K` скопирует абсолютный путь к файлу. Вставьте его в избранную заметку, предварив ёлочкой в право.



При переходе по ссылке вы получаете несколько отличающийся список заметок.



Заметка, на которую ссылаются другие записи, всегда будет сверху³⁵. Удобно это тем, что вы можете перейти по ссылке, просто нажав RET два раза. Еще один аргумент в пользу подобных ссылок в том, что именно они используются в `howm`, когда вы создаете цепочку заметок (смотри стр. 18). Конечно, просто даты в списке малоинформативны. Чтобы отобразить заголовки, нажмите `T`.

На самом деле абсолютный путь к файлу позволяет использовать стандартную для Emacs ссылочную систему. Это обеспечивает мгновенный переход из одной заметки в другую³⁶.

Вместо ёлочки-вправо напечатайте префикс `file://`

³⁵ В случае со ссылкой на заголовок список формируется иначе и адресная заметка может оказаться посередине или в конце.

³⁶ Правда при таком переходе `howm` не распознает заметку как родную (если только она уже не открыта в буфере). А это значит, что и синтаксис `howm` работать не будет. Вам придётся дополнительно его активировать: `M-x howm-mode`

Лучше, как мне кажется, привыкнуть к ссылочной системе howm (ссылка-список-переход). Она обладает целым рядом преимуществ при работе с заметками³⁷.

Продумайте также, когда использовать в качестве ссылок нумерическое имя файла, а когда заголовок. Желательно, однако, сохранять последовательность. При переходе по ссылке с нумерическим именем вы получите список всех заметок, которые содержат такую же ссылку. Однако howm не будет показывать ссылки на заголовок. Обратное тоже справедливо.

Преимущество ссылок на заголовок или любое другое словосочетание в тексте в том, что вы видите неучтенные, но потенциально важные связи. Минус – в смысловом шуме, если результатов в поисковой выдаче слишком много.



Используя нумерическое имя файла вы вполне можете организовать обратные ссылки там, где они необходимы. Для этого просто разместите в заметке ссылку на нее саму.

1. Копируйте имя файла командой `C-u C-c , K`
2. В той же заметке наберите `>>>` и нажмите `C-y`
3. Пройдя по ссылке, вы откроете список заметок, которые содержат нумерическое имя текущего файла (ссылаются на него).

Вы можете также быстро проверить наличие обратных ссылок, не внося в заметку изменений. Это даже проще. Используйте последовательность команд: `C-u C-c , K C-c , s C-y RET`

Если необходимо сделать ссылку на какой-либо файл за пределами директории howm, можно использовать уже упоминавшийся префикс `file://`

Дальше необходимо указать путь к файлу: `file://~/папка/нетленка.org`

В howm есть и более удобный способ с ёлочкой: `>>> ~/папка/нетленка.org`

В первом случае переход по ссылке сразу перебросит вас в адресный документ. Во втором случае вы получите традиционный для howm двухоконный фрейм. В нижнем окне будет пред просмотр файла, в верхнем – список ссылающихся на него заметок. Это обеспечивает как большую информативность, так и гибкость навигации. Вы можете работать с любым документом, не покидая при этом howm.

³⁷ Прежде всего, это возможность напоминать себе о других траекториях в ссылочной системе.

Без проблем внутри Emacs открываются любые plain text файлы (типа txt, md, org). Картинки тоже открываются. Для более замысловатых форматов типа Docx, Pdf, DJVU могут потребоваться дополнительные плагины и настройки Emacs.

Возможность выстроить ссылочную систему, объединяющую разные документы в разных директориях, - это огромное преимущество. Однако имейте в виду, что переместив какой-либо файл в другую папку или переименовав его, вы поломаете ссылку³⁸.

Howt также распознает обычные ссылки на интернет страницы. Любая ссылка, скопированная из адресной строки браузера, будет рабочей³⁹.

Что если перед вами противоположная задача? Как сделать ссылку на карточку howt из какого-то другого документа на вашем компьютере?

Например, я не храню в картотеке конспекты книг. Однако зачастую мне нужно упомянуть в конспекте какие-то идеи из моих заметок. Ссылка крайне желательна.

Я покажу, как создать внешнюю ссылку на карточку howt, на примере документа Org. Причина такого выбора проста: Org-mode – это один из лучших инструментов для работы с длинными текстами⁴⁰.



1. Откройте карточку, на которую вы хотите сослаться.
2. Откройте еще один фрейм Emacs – `C-x 5 2`
3. Откройте в новом фрейме нужный файл Org.
4. Вернитесь обратно к заметке – `C-x 5 o`
5. Разместите курсор на пустой строке и нажмите `C-c l`
6. Вернитесь к документу Org
7. Вставьте ссылку – `C-c C-l`

³⁸ Лучше всего выработать для себя максимально простую и понятную систему именования файлов и создания директорий. Универсальных решений здесь нет, но чем меньше степеней вложенности, тем лучше.

³⁹ Emacs позволяет просматривать интернет-страницы. Однако для этого необходимы дополнительные настройки. По умолчанию ссылка откроется в вашем штатном браузере.

⁴⁰ В разделе «Дополнительные настройки» (стр. 82) можно узнать, как подружить режимы howt и Org.

Я бы не сказал, что система линкования в howt самая эргономичная. Но она вполне функциональна. К ней можно привыкнуть, ее можно адаптировать под свои задачи.

Одно ограничение howt при работе со ссылками сразу бросается в глаза. Здесь нет автозаполнения ссылок. Придется найти обе заметки, которые вы хотите залинковать, и скопировать адрес одной в другую. Избалованным автоматизацией людям это может показаться ужасно медленным процессом.

На самом деле, как только вы поймете алгоритм, никаких затруднений процесс вызывать не будет. Удобнее всего воспользоваться уже имеющимся в howt потенциалом списков.

Любая заметка, в которую внесены изменения, перемещается наверх. Это дает возможность расположить необходимые записи рядом друг с другом.



1. Откройте заметку А. Нажмите space и backspace. Сохраните «изменения» – `C-x C-s`
2. Найдите заметку В с помощью команды `C-c , s`
3. Скопируйте имя файла или заголовок
4. Вызовите список недавних правок – `C-c , l`
5. Заметка А будет сверху. Перейдите в нее и вставьте ссылку.

Этот лайфхак относится к старым заметкам, которые вы решили залинковать. Если вы работаете с заметкой А, шансы велики, что первый пункт с искусственными изменениями лишний.

По мере разрастания записей, в одной заметке может оказаться сразу несколько ссылок, причем в разных параграфах. Перемещения между ними можно осуществлять очень быстро

`C-c , n` – Следующая ссылка.

`C-c , p` – Предыдущая ссылка.

В следующем разделе мы поговорим о функции ёлочки-влево.

КЛЮЧЕВЫЕ СЛОВА В HOWM

Какую роль играет ёлочка, повернутая влево?

Она тоже создает ссылку, но с другими свойствами. Можно назвать это «ссылка-сюда», а предыдущий тип – «ссылка-отсюда».

Ёлочка-вправо >>> символизирует переход из текущей заметки в другую, то есть ссылка ведет отсюда.

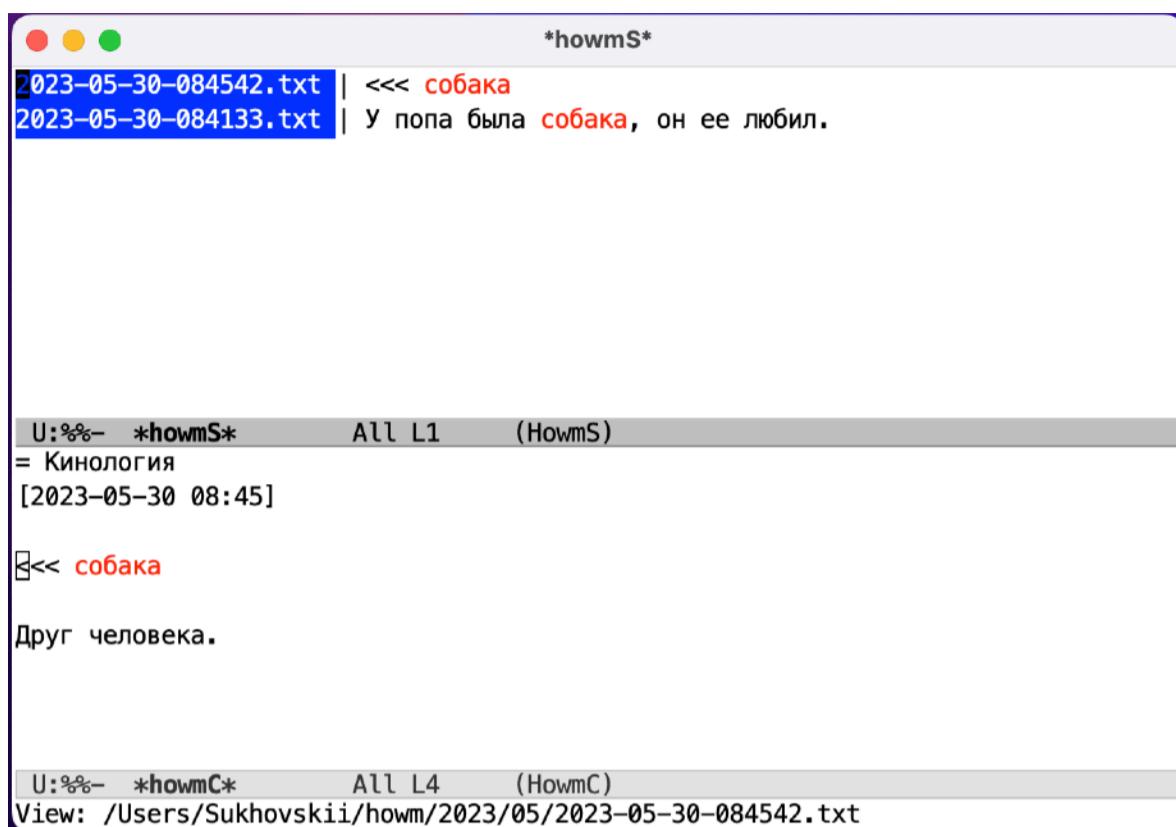
Ёлочка-влево <<< имеет прямо противоположный смысл. Вы не хотите никуда переходить. Вы хотите, чтобы все другие заметки с ключевым словом ссылались сюда.

Также как и в случае с ёлочкой-вправо, все что расположено после <<< будет расцениваться как ссылка.

Давайте посмотрим, что это означает на практике.



Если вы разместите курсор на ссылке в Заметке А и нажмете RET, вас перебросит в стандартный для howm двухоконный режим со списком.



Еще раз напомню, что ссылка – это триггер поискового запроса, а промежуточный результат – список заметок, соответствующих критериям⁴¹.

Поскольку заметка с ключевым словом всегда будет сверху списка, остается нажать RET, чтобы в нее перейти⁴².

Два важных примечания.

1. Ссылка-отсюда активируется сразу после того, как вы ее набрали. С ссылкой-сюда такого не происходит. Вам нужно сохранить заметку, чтобы она заработала – **C-x C-s**
2. Когда вы набрали ключевое слово в новой заметке, оно не преобразуется в ссылку динамически. Howm нужно сообщить об изменениях в тексте. Для этого можно сохранить заметку или нажать команду **C-c , r**. Данную команду можно также запустить из начального меню howm (там это клавиша **r**).
3. Ссылка-сюда чувствительна к регистру. Если вы использовали слово с заглавной буквы, только такие слова и будут преобразованы в ссылки в других заметках. Однако поиск, который инициируется при нажатии на ссылку, регистр игнорирует. В поисковой выдаче (списке) будут слова и с заглавной, и со строчной буквы.

⁴¹ Подробнее о поиске в следующей главе.

⁴² Разместив курсор на подчеркнутом слове и нажав RET два раза, вы сразу перемещаетесь в адресную заметку.

Ссылка-сюда может оказаться полезной при создании разного рода указателей (тематического, предметного, именного).

Например, вы можете создать карточку, посвященную какому-либо человеку, и разместить там ёлочку-влево (<<< Диоген). Упоминание этого человека в других записях теперь будет ссылкой.

В русском языке неизбежно возникает проблема падежный окончаний. Если вы назначили ссылку <<< собака, как в предыдущем примере, howt проигнорирует варианты: собаки, собакой, собаку, собаке.

С другой стороны, в случае со ссылкой <<< Диоген, в список будут включены все варианты: Диогена, Диогену и т.д.

Самый простой способ обойти проблему падежный окончаний, указывать слово без них – <<< собак

Ссылки-сюда в howt обладают еще одной интересной и полезной особенностью. Вы можете указать alias для ключевого слова, проще говоря, список синонимов или близких по смыслу понятий.

Выглядит это так: <<< эвристика <<< серендипность <<< интуиция

После того, как вы создали подобную цепочку, все указанные слова будут адресовать к заметке со ссылкой-сюда. При переходе по ссылке со слова «серендипность» вы увидите список, в который будут включены заметки со словами «эвристика» и/или «интуиция».

Ссылки-сюда могут выполнять роль схожую с тегами.



Что если вам нужны не все упоминания какого-то слова, а лишь конкретные заметки? Используйте какой-нибудь значок перед словом, и тогда howt будет искать только указанный вариант.

Например, ссылка-сюда <<< :интуиция: будет игнорировать обычное упоминание слова «интуиция» в других заметках и сработает только, если вы разместите где-то тег :интуиция:

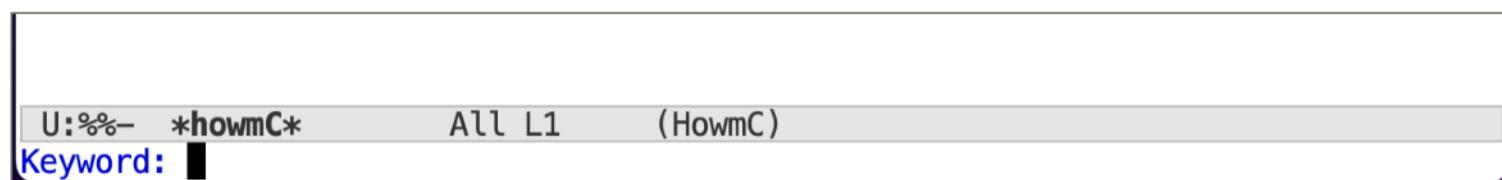
Такой тип тегов используется в Org-mode, в терминальных приложениях типа Vimwiki и т.д.

Если вы дополнительно работаете в приложениях, где теги обозначаются через значок решетки, используйте его: <<< #интуиция

Конечно, помнить ключевые слова и заново их вводить – занятие муторное. На определенном этапе работы их станет достаточно много.

В Howm есть специальная команда для вызова всех ключевых слов – `C-c , i`

Команда дает вам доступ к списку всех внесенных в базу ссылок. Для этого howm вызывает диалог минибуфера:



Можно набрать несколько первых букв искомого слова и затем нажать клавишу табуляции (в howm, как и повсюду в Emacs работает автодополнение). Если вы хотите увидеть полный список ключевых слов, нажмите на табуляцию сразу⁴³.

Тут стоит пояснить, что howm вносит все ссылки сюда в особый файл, который называется `.howm-keys`



Если вы ничего не меняли в настройках, он располагается прямо в домашней директории. Хотя лучше его разместить в папке howm⁴⁴.

Вы, кстати, можете редактировать этот файл напрямую. Слово, внесенное в этот список, оказывается под наблюдением howm, даже если ни в одной заметке оно не указано в ссылке-сюда.

Последнее может быть удобно как раз в случае с тегами. Переход по ссылке-тегу открывает список тегированных заметок, которые рассортированы в порядке новизны правок.

⁴³ Эта и многие другие функции лучше работают, если у вас установлен дополнительный пакет Vertico.

⁴⁴ См. раздел «Дополнительные настройки», стр. 79.



Следует иметь в виду, что простое удаление ёлочки-влево не деактивирует ссылки в других заметках. Они будут по-прежнему запускать поиск. Просто теперь список будет без адресной заметки сверху.

Чтобы убрать «ссылку-сюда» полностью, необходимо удалить соответствующее слово в файле .howm-keys

Я все еще не упомянул дополнительный тип ссылок, который можно использовать в howm. Это знакомые многим по wiki и markdown ссылки в квадратных скобках. Здесь они ведут себя специфично.

The screenshot shows two windows of the Howm application. The top window is titled '2023-05-30-084133.txt' and contains the following text:

```
= Заметка А  
[2023-05-30 08:41]  
  
У попа была собака, он ее любил.  
[[Пушкин]]
```

To the right of this window is a callout box with the following text:

Создайте имя новой заметки и заключите его в квадратные скобки. Это активирует ссылку.

Below this is another callout box with the following text:

Разместите курсор на ссылке и нажмите RET. Это создаст новую заметку.

The bottom window is titled '2023-06-04-134953.txt' and contains the following text:

```
= <<< Пушкин  
[2023-06-04 13:49] >>> ~/howm/2023/05/2023-05-30-084133.txt
```

To the left of this window is a callout box with the following text:

Ссылка на Заметку А.
Howm добавляет такие ссылки всякий раз, когда одна заметка создается из другой.

Annotations in the bottom-left note area include:

- A red arrow points from the word 'Заголовок' in the note to the title bar of the note window.
- A red arrow points from the word 'Пушкин' in the note to the link in the note window.
- A red arrow points from the word 'Файл' in the note to the file path in the note window.

Annotations in the bottom-right note area include:

- A red arrow points from the link in the note to the link in the note window.

Достоинства ссылок в квадратных скобках:

1. Вы создаете ссылку, `howt` при переходе создает заметку (если ее еще не существует).
2. Ссылка не требует после себя пустой строки, как в случае со ссылками ёлочками. В итоге текст выглядит более компактно.

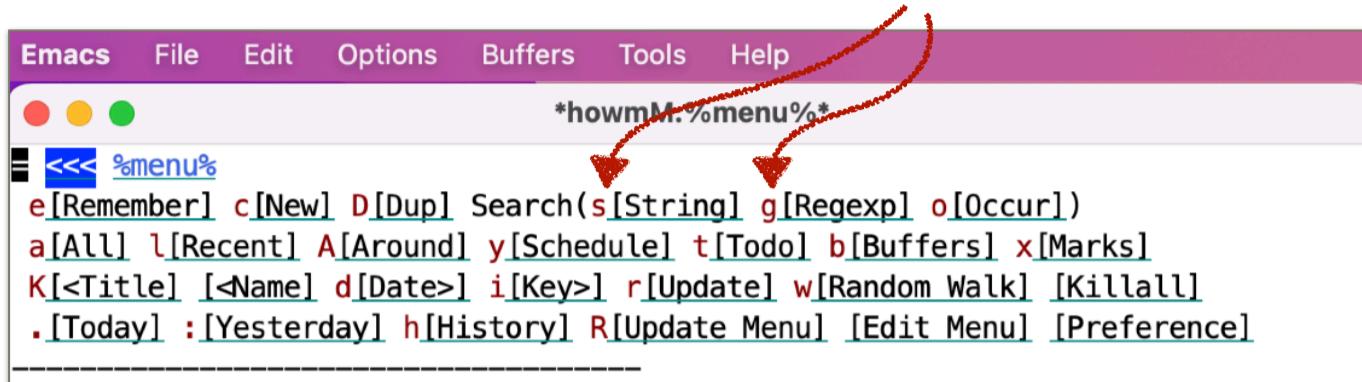
Недостатки подобных ссылок:

1. Подобные ссылки хорошо работают лишь с заголовками заметки. Однако нумерические ссылки надежнее и универсальнее.
2. Заголовки заметок будут скапливаться в файле `.howt-keys`, который разумно зарезервировать под указатели и теги. Не знаю, отразится ли это на быстродействии системы, однако, поиск по ключевыми словам будет точно затруднен.

ФУНКЦИИ ПОИСКА

В этом разделе мы рассмотрим функции поиска в howm. Здесь множество полезных трюков, которые позволяют эффективно ориентироваться на местности.

Начальная страница howm содержит необходимые подсказки.



Чтобы инициировать поиск, переместите курсор на соответствующий пункт меню и нажмите RET или сразу нажмите клавишу, которая связана с командой.

Если вы уже работаете с заметками, нужды возвращаться к начальной странице howm нет⁴⁵. Поиск вызывается клавиатурными сочетаниями: **C-c , s** и **C-c , g**

Команда **C-c , s** позволяет искать текст по части слова, слову или словосочетанию.

Кроме того, благодаря этой команде вы получаете доступ к списку терминов, внесенных в базу howm (файл .howm-keys). Это те самые ссылки-сюда. Если после вызова команды вы нажмете на клавишу табуляции, то получите их полный список. Наберите первые буквы нужного понятия и нажмите клавишу табуляции снова для автозаполнения⁴⁶.

В данном случае результат будет таким же, как если бы вы кликнули по ссылке-сюда в какой-либо заметке (помните, что переход по ссылке в howm - это просто поисковый запрос?).

Команда **C-c , g** также позволяет искать текст по части слова, слову, фразе. Но здесь функционал поиска расширен благодаря регулярным выражениям.

Например, написав: **Бяки \! Буки** вы получите все заметки, где встречаются эти слова. Это называется логическим «ИЛИ».

⁴⁵ Вы также можете перейти к поиску сразу после запуска Emacs.

⁴⁶ Еще раз отмечу, что с пакетом Vertico это делать удобнее. Есть, правда, один побочный эффект. Vertico слишком услужлив. Предположим, в список ключевых слов у вас включено понятие «филология». Если вы решите найти записи со словом «филолог», ничего не получится. Буквы в вашем поисковом запросе полностью совпали со словом внесенным в список, и Vertico будет автозаполнять недостающее. В подобной ситуации используйте команду - **C-c , g**

Могут быть и более замысловатые команды. Если вы хотите их освоить, обратитесь к синтаксису регулярных выражений (regexp). Для работы в howm эти штуки являются факультативными.

В случае простого поискового запроса подойдет любая из двух указанных команд.

Рассмотрим теперь как ведет себя howm в процессе поиска.

Запросите любое слово и howm выведет вам список заметок, которые его содержат (поиск нечувствителен к регистру).

The screenshot shows a window titled '*howmS*' containing a list of file names on the left and their contents on the right. The file names are: 2022-11-07-124317.txt, 2023-06-05-150627.txt, 2022-08-15-143955.txt, 2022-10-14-143915.txt, 2022-07-13-220245.txt, 2022-07-14-115335.txt, 2022-08-29-093519.txt, and 2022-07-13-215823.txt. The file 2022-11-07-124317.txt is currently selected. The right pane displays the content of this file, which includes several occurrences of the word 'компьютер' (computer) highlighted in red. A cursor is visible in the text area. The status bar at the bottom shows 'U:%%- *howmS*' and 'Top L1 (HowmS)'.

Поисковая выдача разделена на два окна. В верхнем список заметок и искомым словом. В нижнем предпросмотр заметки, на которой находится курсор. Это позволяет быстро пролистать записи.

Нам сейчас важнее верхнее окно. Здесь у нас синий столбик с нумерическими именами заметок и справа от него выдержки из заметок, где красным подсвеченено искомое слово.

Если в заметке слово встречается несколько раз, будут указаны все случаи, причем каждый отдельной строкой. В левом столбце повторные упоминания легко заметить. Это пустая темно синяя графа. Стало быть, упоминание в той же заметке, нумерическое имя которой указано выше.

Вы можете свернуть эти дополнительные строки, нажав клавишу **и**. В списке останутся выдержки с первым упоминанием слова в заметке. Чтобы вернуть подробную сводку, нажмите на **и** повторно.

Иногда легче сориентироваться в материале не по выдержкам, а по заголовкам заметок. Нажмите клавишу **T**, и **howt** будет отображать справа от нумерического имени заголовки. Так у вас появляется дополнительная подсказка, позволяющая быстрее найти нужный материал. Чтобы вернуться к списку с выдержками, опять нажмите клавишу **T**.

Поисковая выдача со списком заметок - это уже хорошее начало для работы. Однако в **howt** вы можете прибегнуть к дополнительному уточнению и реорганизации материала.

Иногда поиск по одному слову создает слишком большой список заметок. Например, вам не нужны все заметки по философии. Однако вас интересуют записи, где упоминается и философия, и Аристотель. Это называется логическим «И».

В **howt** такой запрос приходится выполнять последовательно. Сначала поиск по одному слову, а потом сужающий поиск по другому.

В этом есть свой плюс. Вы можете осуществить один раунд поиска, а уж потом решить, как его сузить. Операцию можно повторять снова и снова.

Данная функция связана на клавишу **G**. После поиска по первому слову, нажмите **G** и введите второе слово.

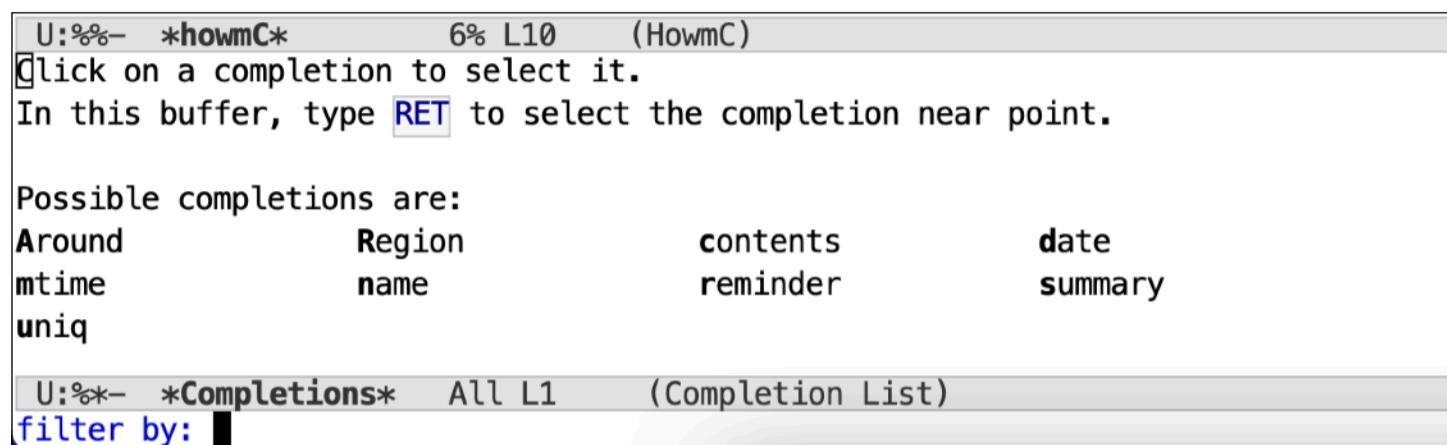
Вы можете отсеять заметки из поисковой выдачи и другим способом. Об этом будет следующая глава.

Осталось упомянуть еще одну полезную функцию. Иногда вам приходится повторять один и тот же поисковый запрос по несколько раз. **Howt** хранит историю поисковых запросов в отдельном файле. Список можно вызвать командой – **C-c , h**

ФУНКЦИЯ ФИЛЬТРАЦИИ

Нажмите клавишу **f**. Nowt откроет диалоговое окно фильтрации заметок. Эта функцию можно использовать не только в контексте поиска, так как она работает с любыми списками (например с полным списком заметок).

Тут масса вариантов, помогающих вам сфокусироваться на отдельном сегменте. Нажмите Tab, чтобы увидеть полный перечень.



Рассмотрим открывшиеся перед нами возможности в порядке важности.

name

Речь о нумерическом имени файла в синем столбце. Указав, например, 2023-03, вы получите список заметок, созданных в марте. Но может быть, вам нужны заметки, которые вы создали в прошлый понедельник. Тогда добавьте к году и месяцу дату.

date

Фильтр, похожий на предыдущий. Однако здесь вы можете задать диапазон. Скажем, вас интересуют заметки, созданные с декабря 2022 по февраль 2023.

mtime

Здесь тоже указывается временной диапазон, но на выходе у вас будет список заметок, с которыми вы работали в указанный период (то есть, не только создали, но и изменили).

summary

Справа от синего столбца с нумерическим именем файла находится выдержка с подсвеченным поисковым словом. В выдержку включена строка, в которой данное слово находится.

Вы можете сузить поисковую выдачу на основании этой информации. Например, вас интересуют случаи, где в пределах одного предложения встречаются два определенных слова (Скажем, Сократ и Платон).

`summary` можно переключить с отображения выдержки на отображение заголовков (если вы из создали). Для этого нажмите клавишу `T` (вернуться обратно можно, нажав ее повторно).

Такая комбинация позволяет очень быстро обнаружить необходимую заметку. Вы помните из нее какое-то слово - первый поиск. Дальше переключитесь к отображению заголовков. Если их много, нажмите `f`, выберите `summary` и укажите другое слово, чтобы сузить список.

contents

С помощью этой команды `howt` осуществит поиск в уже отобранных заметках. Эта команда выполняет аналогичную функцию, что и клавиша `G`. Полезно, когда у вас есть список заметок с искомым словом, а потом решили сузить его дополнительно.

reminder

Команда, позволяющая отфильтровать списки задач, если вы их в `howt` используете (об управлении делами мы поговорим в одном из следующих разделов).

Region

В Emacs вы можете выделить текст или элементы списка, нажав на `C-space` и затем перемещая курсор. Выделите нужные вам заметки, затем нажмите `f` и выберите опцию `Region`. `Howt` уберет все остальное.

Around

Фильтр, который позволяет взять любую заметку (переместив на нее курсор) и запросить список других заметок, с которыми вы работали до и после указанной. Полезно, когда нужно вспомнить контекст.

Вы можете указать конкретное число заметок в контексте. Например, команда `C-u 10 f` плюс `Around` выведет на экран контекст из 10 заметок до и 10 заметок после активной.

uniq

Команда, дублирующая функцию, связанную на клавишу `u`.

* * *

Сократить список заметок при поиске можно не только благодаря логическому «И», но и благодаря логическому «НЕ».

Например, из поисковой выдачи по теме философия вам захотелось убрать заметки о Кьеркегоре.

Наберите `C-u f`

Howt спросит по какому критерию осуществить отсев. У вас на выбор:

`name`

Нумерическое имя файла (можно вычеркнуть одну определенную заметку, а можно целиком какой-то месяц или год).

`date`

Тот же отсев по дате, но можно указать какой диапазон дат вы хотите убрать.

`mtime`

Отсев по дате правок. Например, вы хотите убрать из списка заметки по философии, с которыми работали год назад.

`summary`

Можно вычеркнуть из списка заметки на основании выдержки слева от нумерического имени файла или на основании заголовка (переключитесь клавишей T).

`contents`

Вычеркнуть заметки на основании содержания.

Функции `Around`, `Region`, `reminder`, `uniq` в данном случае бесполезны, так что не обращайте на них внимания.

К сожалению, в howt нет возможности сохранять все эти хитрые комбинации фильтров для повторного использования.

То есть, с одной стороны жаль, а с другой - я понимаю, что насоздавал бы себе кучу фильтров, в которых сам бы стал путаться.

УДАЛЕНИЕ ЗАМЕТОК

Иногда какую-то заметку нужно удалить. Делать это стоит только в крайних случаях. Например, вы обнаружили, что запись А полный дубликат записи В.

В `howm` нет быстрого способа удаления (одним кликом). Может быть, и к лучшему. Ампутация текстов требует сознательности.

Сначала проверьте, не связана ли заметка, от которой вы хотите избавиться, с другими записями. Неосторожное удаление может привести к образованию битых ссылок в системе.

Самый простой способ проверки таков:

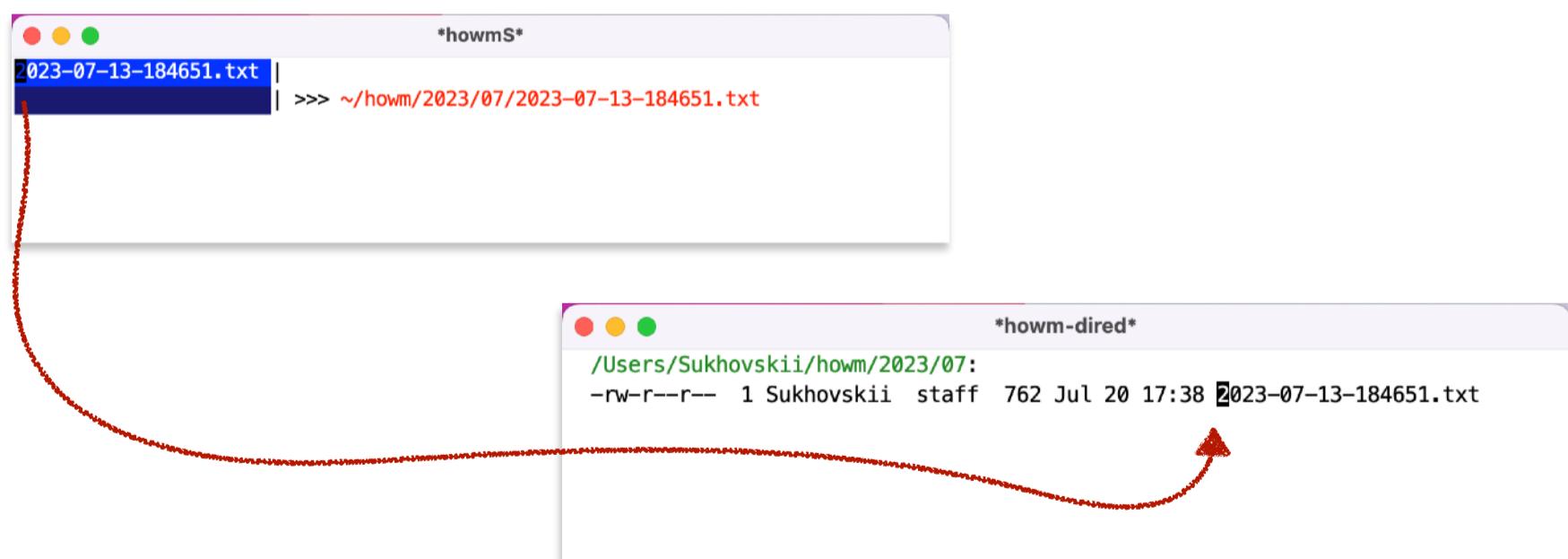
Используйте трюк со ссылкой заметки на саму себя⁴⁷. Для этого вам потребуется последовательность команд: `C-u C-c , K C-c , S C-y RET`

В списке должна оказаться всего одна заметка.

Если вместо нумерических ссылок вы используете ссылки на заголовки, процедура почти идентична. Копирование заголовков осуществляется командой `C-c , K`

Как только вы убедились, что удаление заметки пройдет для картотеки безвредно, можно перейти к хирургическому вмешательству.

Для этого необходимо перейти в режим `Dired`. Если курсор в списке на обреченной заметке, нажмите `X`. Экран изменится, однако, курсор будет указывать на ту же заметку.



⁴⁷ Мы уже обсуждали, как это сделать на примере обратных ссылок. См. стр. 32

В новом окне вам доступны все стандартные для режима `Dired` манипуляции с файлами⁴⁸.

Нажмите `v`, чтобы заглянуть в файл, если нужна перестраховка. Нажмите `q` для возврата в режим списка `Dired`.

Чтобы удалить файл нажмите `D`. `Emacs` переспросит вас, осознаете ли вы, что действие безвозвратно, после чего удалит файл⁴⁹.

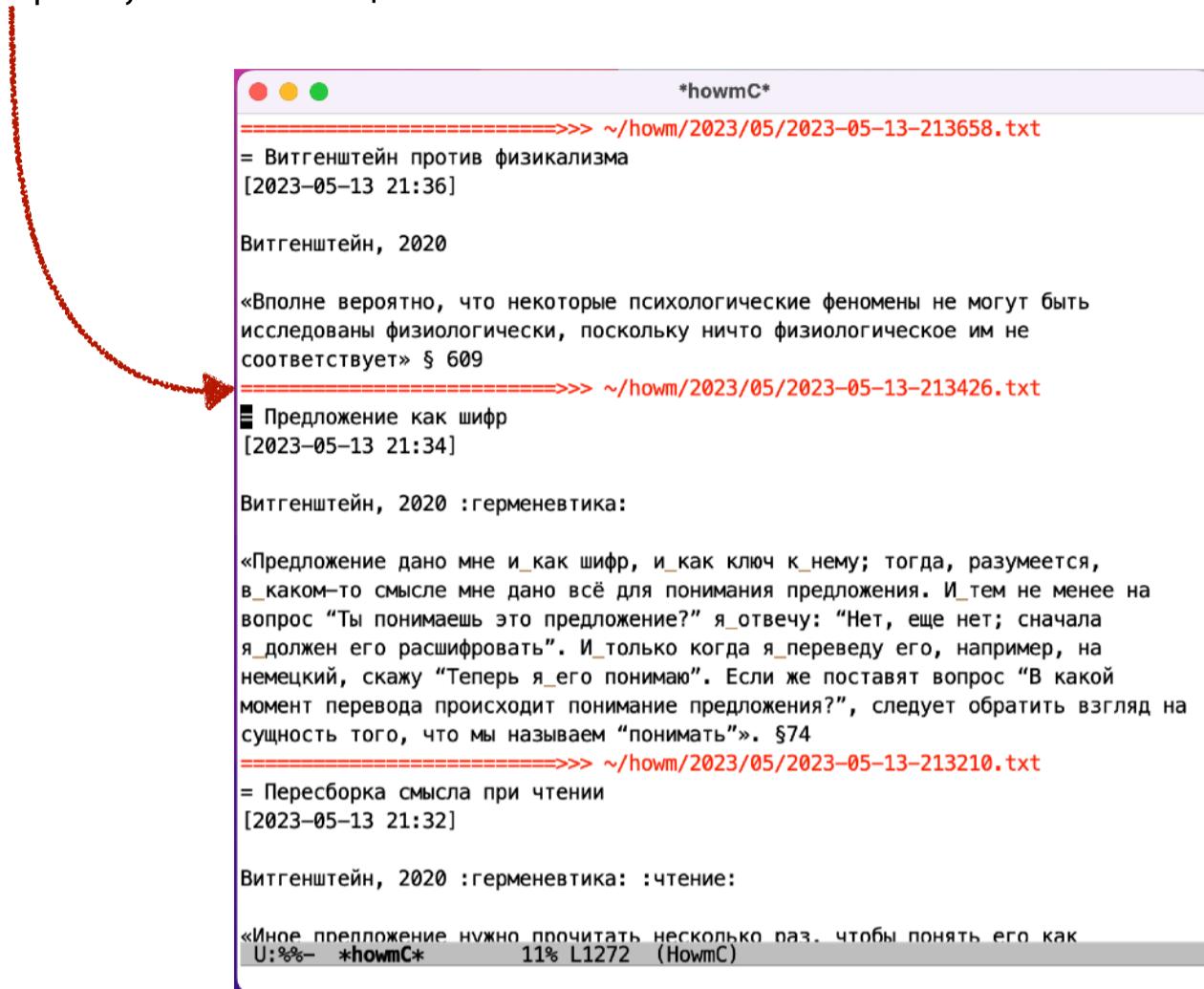
Вы можете выбрать сразу несколько файлов для удаления. Для этого нажимайте последовательно клавишу `d`. `Emacs` пометит файлы. Для подтверждения операции следует нажать `x`. Если вы решили снять метку с какого-либо файла, переместите курсор нужную строку и нажмите `u`.

⁴⁸ Как здесь, так и в других случаях, знакомство с документацией `Emacs` существенно расширит свободу ваших действий.

⁴⁹ Файл не попадает в корзину, а полностью стирается из системы, так что будьте осторожнее!

СШИВАНИЕ ЗАМЕТОК

В howm возможен режим сводного чтения заметок. Они будут сшиты в единый свиток с маркерами, обозначающими начало новой записи.



Режим сводного чтения вызывается клавишами **0** или **@** из списка заметок⁵⁰. Выйти из режима можно повторно нажав одну из указанных клавиш.

Эта функция имеет несколько практических применений.

Во-первых, иногда вам просто необходимо преодолеть дискретность картотеки и увидеть записи единым текстом. Так их удобнее читать. Это вполне соответствует девизу howm: «write fragmentarily and read collectively».

Перепрыгивание от заметки к заметке осуществляется клавишами **Tab** и **M-Tab**.

Функция сшивания также полезна при быстром поиске и просмотре заметок. Казуюки Хираока предлагает воспользоваться следующим алгоритмом:

1. Осуществите первоначальный поисковый запрос: **C-c , s**
2. Объедините результат в единый свиток.
3. Воспользуйтесь поиском по тексту, чтобы найти нужное место: **C-s**

⁵⁰ Сшиты будут заметки из вашего текущего списка.

Кроме того, сшивание может служить мостиком между картотекой и писательскими проектами.

Благодаря этой функции легко перейти от этапа сбора заметок к соединению их в единый текст статьи.

Алгоритм работы может быть следующим:

1. Оставьте в записях, которые вы хотите включить в статью, особую метку. Это может быть любое уникальное сочетание букв, например, - @art.
2. Когда записи выбраны, используйте поиск по метке, чтобы сформировать список.
3. Расположите заметки в списке в нужной вам логической последовательности. Для этого придется прибегнуть вот к такому трюку:
 - A. Нажмите клавишу `S`, чтобы вызвать диалог сортировки.
 - B. Используйте опцию `name-match` или `summary-match` для перемещения заметок. Здесь нужно использовать обратную последовательность.
 - i. Найдите первую заметку и переместите ее наверх списка.
 - ii. Найдите вторую заметку и переместите ее наверх списка.
 - iii. Повторите процедуру вплоть до последней заметки.
 - iv. Нажмите `R`, чтобы перевернуть список.
4. Вызовите режим сшитого текста - `0`
5. Выделите текст - `C-x h` и копируйте его `M-w`
6. Откройте текстовый редактор⁵¹ и вставьте туда скопированные фрагменты.

Поскольку заметки теперь обрели вторую жизнь в статье, полезно пометить это в картотеке. Эта практика поможет вам избежать самоповторов. Замените `@art` на «использовано в там-то».

Если заметок не много, можно это сделать и вручную, но в Emacs для таких задач существует замечательная функция поиска и замены по файлам.

⁵¹ Разумеется, если вы используете для работы с длинными текстами Org-mode в Emacs, все эти ухищрения излишни. Можно пропустить пункт №3, а ротацию заметок осуществить уже в Org с помощью заголовков.

Нажмите `C-x d` и локализуйте пространство, в котором вы хотите осуществить операцию (нас интересует директория `~/howt/`).

Нажмите клавишу `t`. Это отметит все файлы в директории. Далее нажмите `Q`.

В минибуфере появится диалог.

Введите `@art` и нажмите RET. Укажите замену - «использовано в статье...» и еще раз нажмите RET.

Emacs уточнит как осуществлять замену. Нажмите клавишу `Y`, чтобы сделать это сразу во всех файлах (поскольку метка уникальна, нам не нужно беспокоиться, что какое-то слово автозаменится по ошибке).

Теперь нужно сохранить файлы, в которые были внесены изменения. Для этого используйте команду `C-x S` и затем нажмите на восклицательный знак (сохранение всех файлов сразу).



Вы можете получить отчет об общем количестве слов, знаков в вашей картотеке, оценить интеллектуальный капитал. Для этого нужно «сшить» все заметки в единый текст.

1. Вызовите общий список заметок - `C-c , a`
2. Нажмите клавишу `0` или `@`
2. Введите команду `M-x count-words` и нажмите RET.

Внизу экрана вы увидите сколько всего понаписано. Цифра, правда, будете не совсем точной, так как `howt` включает в список специальную разметку для деления заметок. Но в целом погрешность не такая большая.

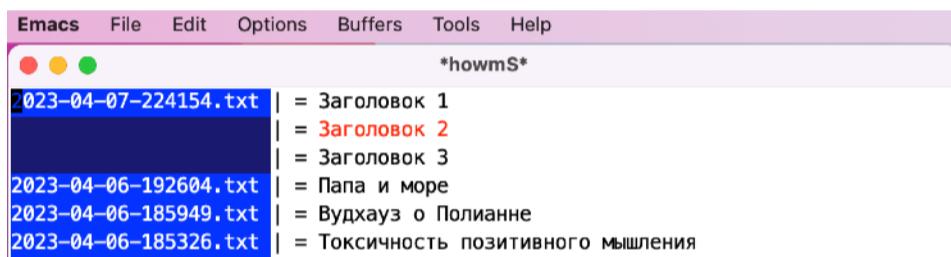
РАБОТА С ДЛИННЫМИ ЗАМЕТКАМИ

В некоторых случаях вам может потребоваться не короткая картотечная запись, а длинный текст. Если удлинения не избежать, разумно хотя бы разделить материал отдельными заголовки.

Каждая заметка в `howm` начинается с заголовка. Однако вы можете разместить в тексте дополнительные разделы:

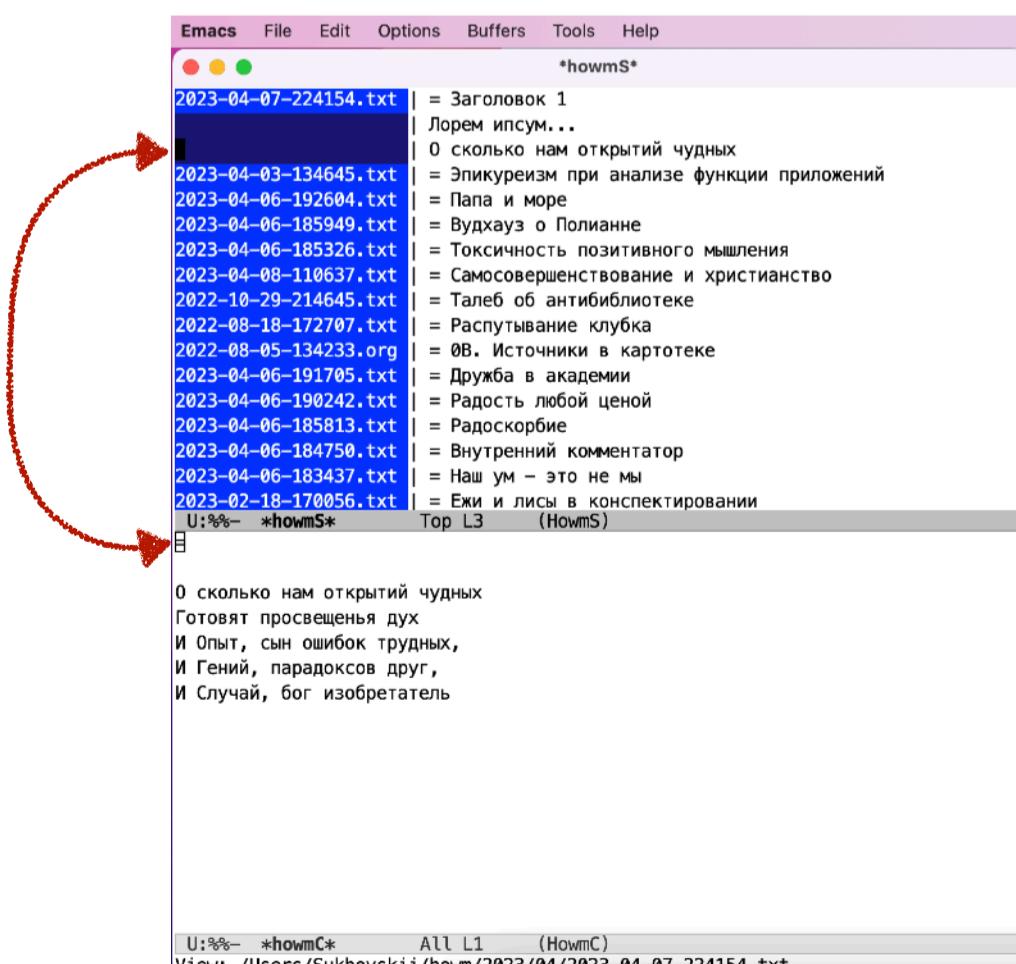
```
= заголовок 1  
= заголовок 2  
= заголовок 3
```

Если вы выведете на экран список заметок, например, с помощью команды `C-c`, а то выглядеть он будет так:

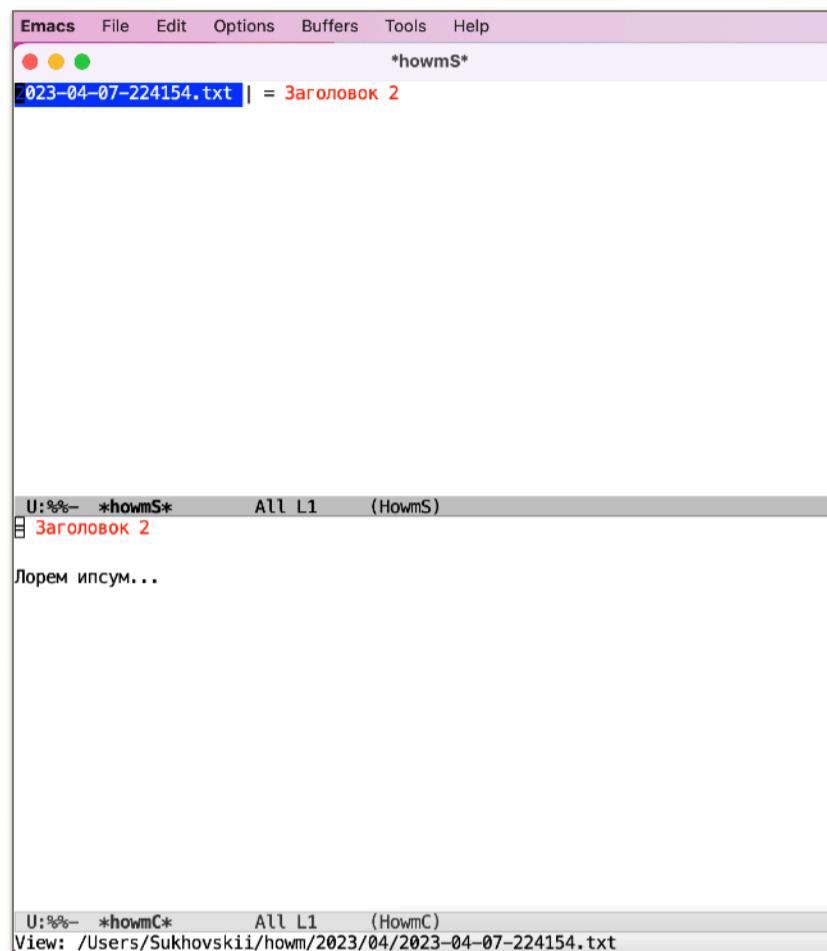


Слева в синей колонке нумерическое имя заметки, а справа все заголовки, которые заметка содержит.

Обратите внимание, что при перемещении курсора, `howm` будет демонстрировать в нижнем окне каждый заголовок, как отдельную заметку (то есть, записи под другими подзаголовками будут не видны, пока вы не перейдете в заметку).



Схожим образом это будет работать при поиске. Вы можете искать заметку по конкретному заголовку. Тогда в верхнем окне поисковой выдачи этот заголовок будет выглядеть как отдельная заметка⁵².



Естественно, нажав на RET и перейдя в заметку, вы обнаружите заголовок в контексте других записей.

Можно использовать знак равно и в качестве разделителя примечаний:

```
= Заголовок заметки
Текст
=
Примечание 1
=
Примечание 2
```

Тогда howm будет в верхнем окне со списком показывать первую строчку примечания. При перемещении курсора по списку в нижнем окне открывается полный текст соответствующего примечания.

Если заметка разрослась в длинный текст со множеством заголовков, возникает необходимость быстрой навигации. Перескакивать от заголовка к заголовку можно, используя следующие команды:

C-c , N → Следующий заголовок.

C-c , P → Предыдущий заголовок.

⁵² В основном меню, в разделе Recent каждый заголовок тоже будет представлен как отдельная заметка. Нумерическое имя, естественно, будет совпадать.

C-c , H → Перейти к первому заголовку.

C-c , L → Перейти к последнему заголовку.

Новый раздел создается знаком равенства в начале строки. Но что если вы хотите создать заголовок с тайм-кодом? Для этого существует отдельная команда **C-c , C**

Новый заголовок подсоединяется к концу текста вне зависимости от того, где находился ваш курсор.

Для длинных заметок полезна функция концентрации на конкретном пункте. В `Howm` для этого не предусмотрено отдельного сочетания клавиш, но сама фича есть.

Команда `M-x howm-narrow-to-memo` скроет все заголовки, кроме текущего (на котором находится курсор)⁵³.

Перед выполнением команды `howm` уточнит ваши намерения. Дело в том, что функция сворачивания/разворачивания заголовков по умолчанию отключена.

В диалоговом окне вы увидите следующие опции:

`y` - активировать режим сворачивания/разворачивания для этой и последующих сессий.

`n` - ничего не активировать и отказаться.

`SPC` - активировать режим единократно. В этом случае `howm` выполнит команду, но режим останется заблокированным.

`!` - активировать все заблокированные команды на время текущей сессии.

Хорошо. Свернули, как теперь восстановить прежний вид?

Для возврата к отображению всех заголовков нажмите **C-x n w**

Вы можете переключаться между режимами скрытия/раскрытия, используя команду `M-x howm-toggle-narrow`

* * *

Если заметка действительно большая, возможно, вам понадобиться осуществить поиск по тексту. Можно прибегнуть к стандартной команде **C-s**

Однако, что если вам хочется получить список всех соответствий?

⁵³ Прелесть Emacs в том, что вы вполне можете назначить собственное сочетание клавиш для вызова команды. Например, напишите в своем `.emacs` следующее:

```
(define-key howm-mode-map (kbd "C-c , =") 'howm-narrow-to-memo)
```

Теперь вы сможете сужать текст до определенного раздела командой **C-c , =**

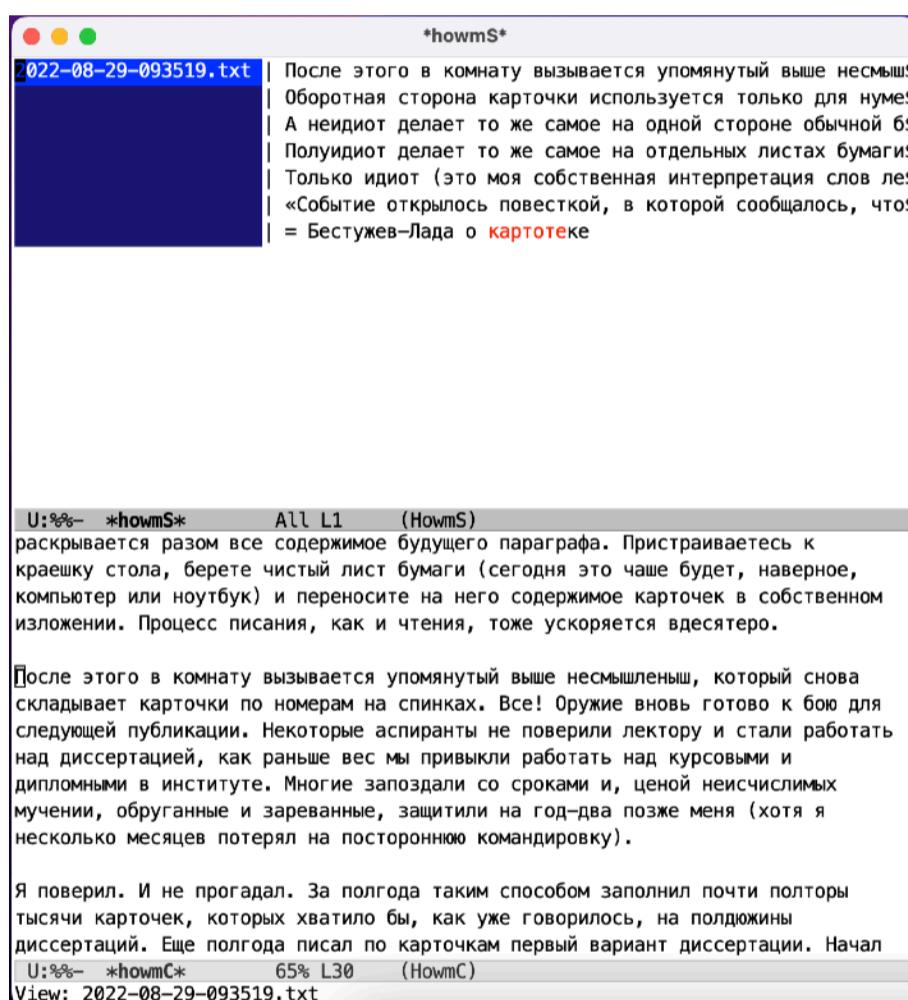
Находясь в заметке, наберите команду `C-c , o`, и вы получите стандартный для `howm` интерфейс: вверху список, внизу текст. Только в данном случае, вы будете перемещаться в рамках одной заметки.

Еще один прием, облегчающий работу с длинными заметками, – это использование меток. Метки – это своего рода закладки, которые вы можете разместить в разных местах текста.

Для того, чтобы создать метку нажмите `C-SPC C-SPC`

Когда метки будут расставлены, вы можете вызвать список командой – `C-c , x`

Перемещение между подобными «закладками» намного быстрее обычного пролистывания⁵⁴.



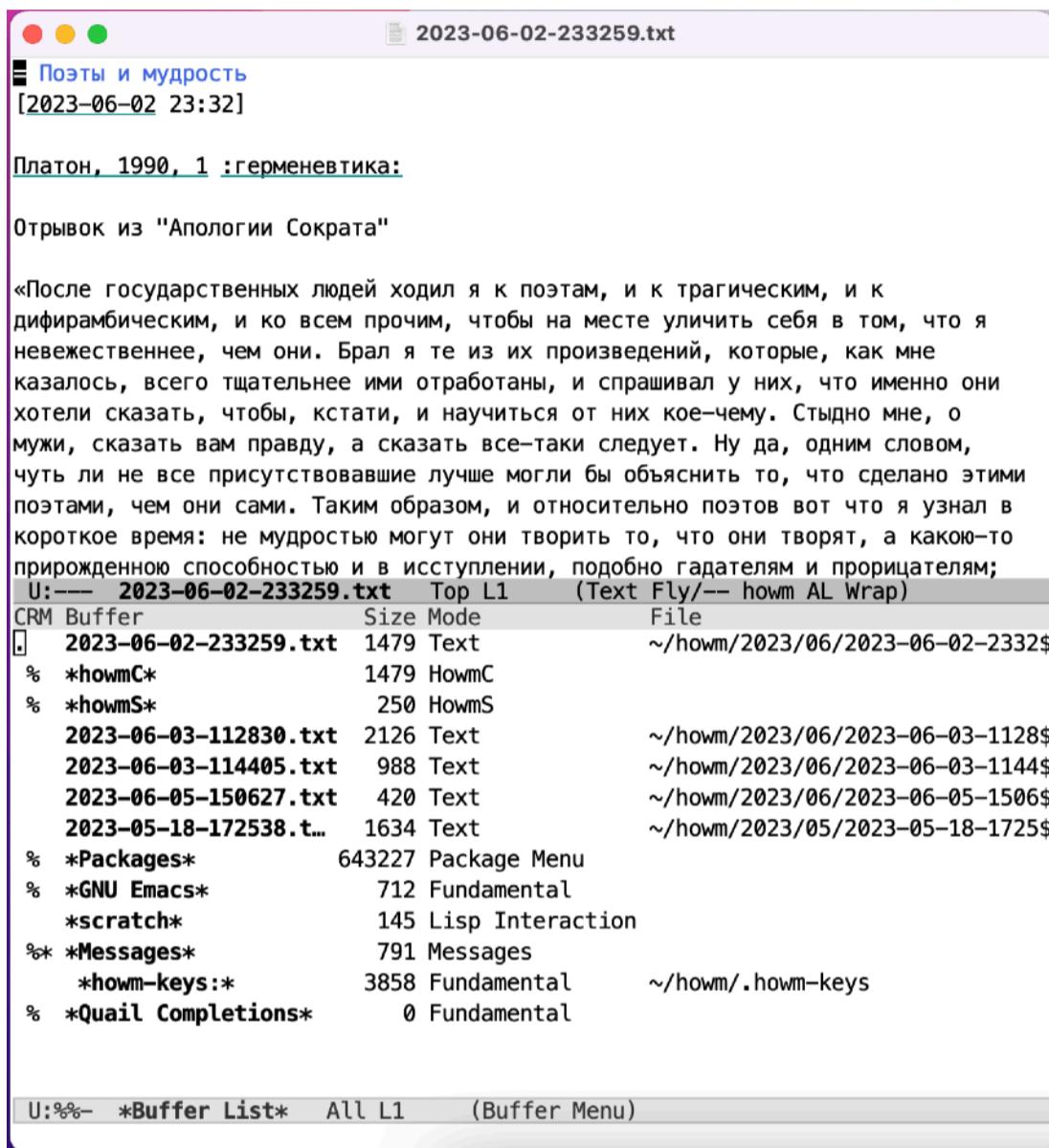
Поскольку ссылка в `howm` – это триггер поискового запроса, вы можете использовать специальные метки, чтобы сослаться на конкретный отрывок в длинном тексте. Например, в одной заметке вы разместите метку – `qq`, а в другой ссылку `>> qq`⁵⁵. При переходе по ссылке `howm` откроет документ в нужном месте.

⁵⁴ На самом деле это список позиций меток в буфере. `Howm` открывает метки активной заметки. Метки в других записях не пропадают, но чтобы их просмотреть нужно перейти в соответствующую заметку.

⁵⁵ Конечно, желательно продумать систему уникальных меток, если вы будете активно использовать этот прием. Q-трюк предложил Мерлин Манн в 2010 году. Он использовал его при работе с заметками в `Simplenote` и `Notational Velocity`.

РАБОТА С БУФЕРАМИ

Emacs хранит каждый открытые вами документы в специальном списке буферов. Стандартная команда Emacs для вызова списка - **C-x C-b**



Список буферов позволяет осуществить много разных задач. Лучше всего обратиться к официальной документации Emacs⁵⁶.

Вы можете переключаться между заметками в howm и другими документами с помощью команды **C-c , SPC**

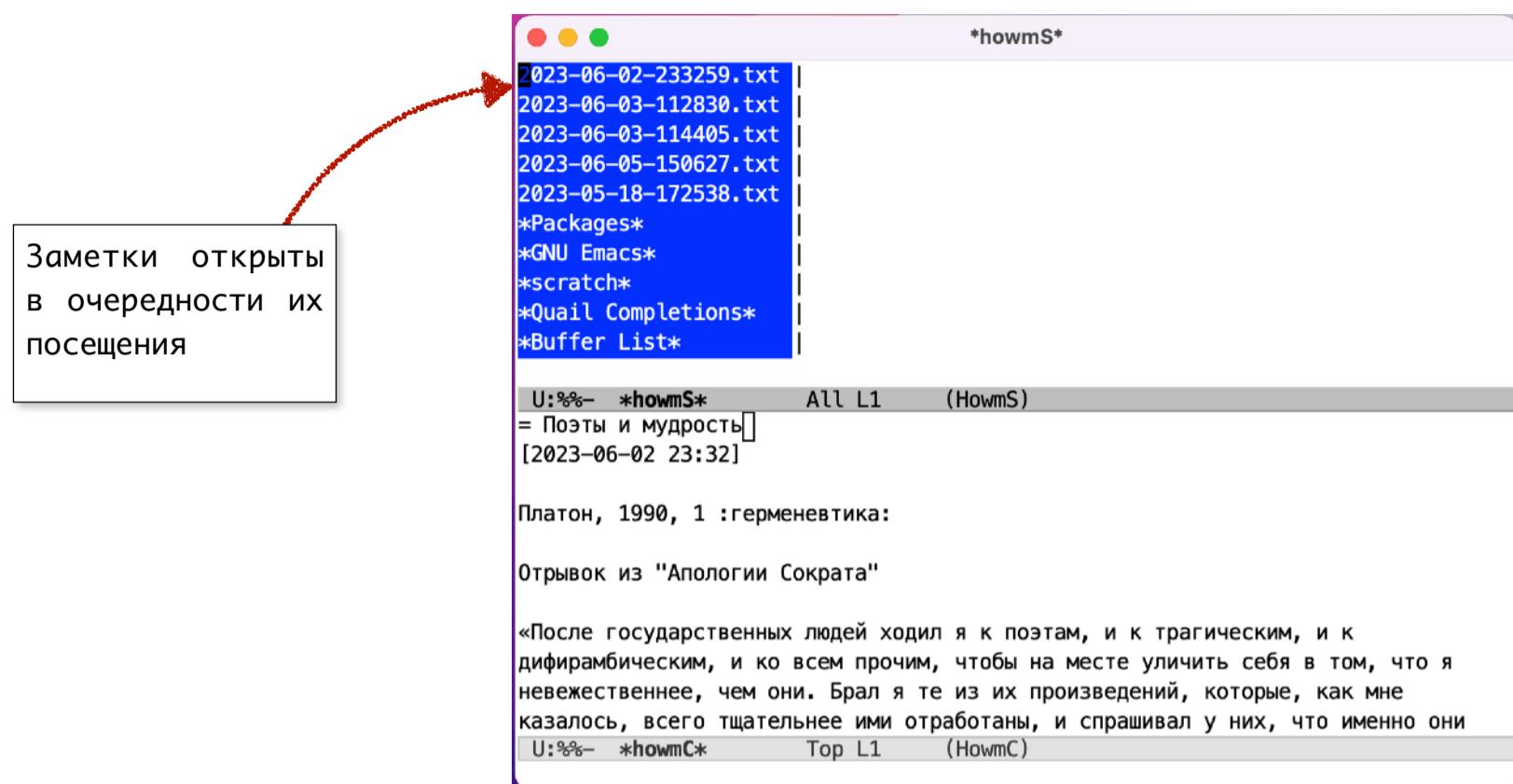
Вас перебросит в последний документ, открытый не в howm-режиме. Набрав команду еще раз, вы вернетесь обратно в заметку⁵⁷.

В howm существует также и собственная команда вызова буферов - **C-c , b**. Преимущества ее в том, что вы получаете традиционный для howm двойной экран. В верхнем окне список буферов, в нижнем предпросмотр.

⁵⁶ См.: https://www.gnu.org/software/emacs/manual/html_node/emacs/Buffers.html

⁵⁷ Не со всеми режимами это работает гладко. Вы сможете переключаться между документами в формате txt. Однако, если вы открыли документ в Org-mode, команды howm в нем не сработают (см. «Дополнительные настройки» стр. 82). Вернуться обратно можно, набрав **C-x ←**

Другие открытые в Emacs документы тоже предпросматриваются.



Эту функцию удобно использовать как «хлебные крошки», то есть, отслеживать ваш маршрут движения по картотеке. Заглянув в список, вы обнаружите файлы, которые вы посещали во время текущей сессии⁵⁸.

В процессе работы над заметками у вас накопится много открытых буферов. Какие-то из них все еще должны быть под рукой, а какие-то уже мешаются под ногами. Как расчистить пространство?

Вы можете набрать команду `C-c , Q`

Это закроет все буферы `howm`, кроме несохраненных (`howm` переспросит, хотите ли вы это сделать). Дальше вас перебросит в последний несохраненный файл `howm` или, если таковых нет, в последний открытый буфер, не связанный с `howm`⁵⁹.

Если вы уже вызвали список буферов – `C-c , b` – достаточно просто нажать `Q`

Команда закрытия буферов доступна также из основного меню – [Killall]

⁵⁸ В списке отображаются нумерические имена файлов. Заголовки были бы понятнее. К счастью, можно внести соответствующие усовершенствования. См. «Дополнительные настройки», стр. 81

⁵⁹ Стандартная команда закрытия буфера – `C-x k` тоже работает.

БЫСТРЫЕ ЗАМЕТКИ

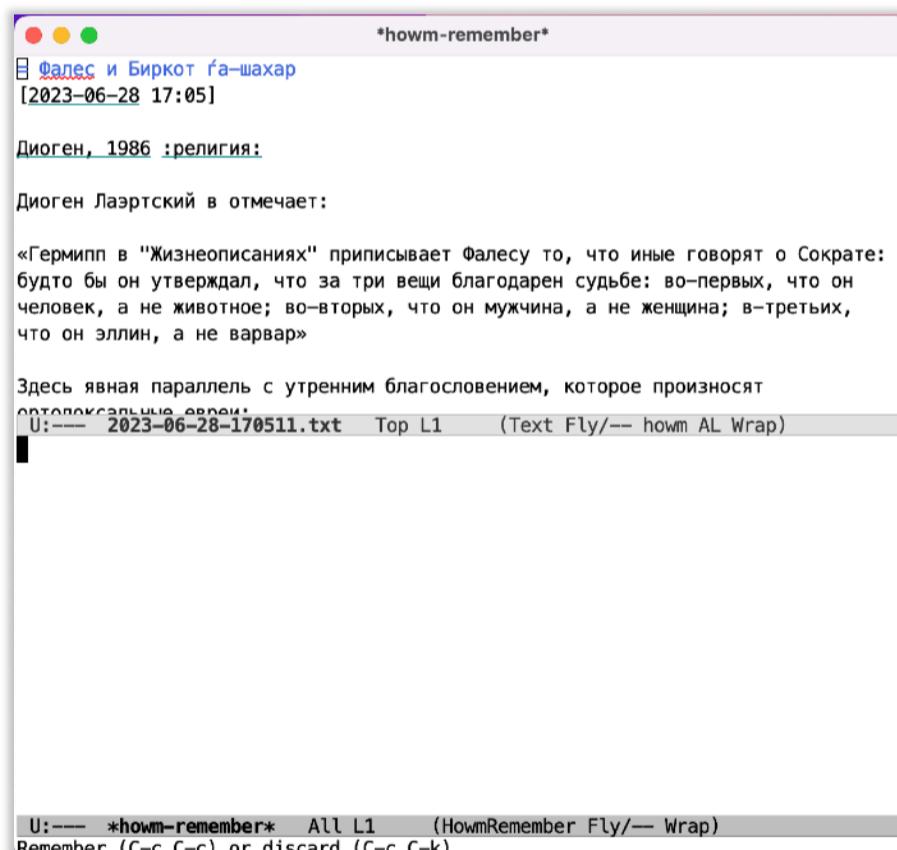
Порой вы пишите одну заметку, а на ум приходит идея для другой. Вы не хотите прерывать состояние потока, но также не хотите упустить инсайт.

В `howm` для таких ситуаций предусмотрена функция быстрых записей. Смысл ее в том, чтобы, не отвлекаясь надолго, оставить себе простую заготовку по теме, которую вы хотите развить позже.

Фактически вы создаёте пустую заметку, в которой есть только заголовок с датой. Заголовок должен послужить подсказкой, о чём собственно вы хотели написать.

В основном меню за данную функцию отвечает кнопка `e[Remember]`. Однако вы можете вызвать буфер быстрой заметки из любого места, набрав команду `C-c , e`

`howm` откроет дополнительное совершенно пустое окно. В минибуфере будут подсказки по командам.



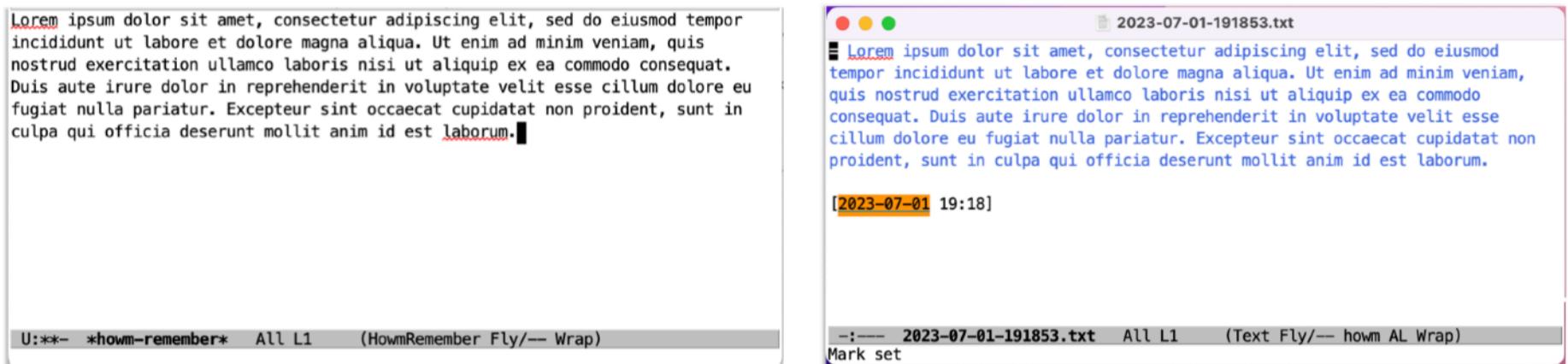
Идеи, попавшие сюда, еще не получили прописки в картотеке. Если вы решите, что они неуместны, нажмите `C-c C-k` и `howm` закроет (убьет) буфер, ничего не сохраняя⁶⁰.

Чтобы переместить запись в картотеку, нажмите `C-c C-c`. `howm` создаст заметку с нумерическим именем и убьет буфер. Вы вернетесь к той заметке, над которой работали до этого.

⁶⁰ При закрытии Emacs содержимое буфера также пропадает. Emacs не будет уточнять, сохранить ли его, как в случае с другими документами. Это похоже на работу специального буфера `*scratch*`.

Когда придет время для ревизии быстрых заметок, вы можете вызвать список недавних файлов – С-с , 1 – или открыть главное меню.

Функция быстрых заметок рассчитана на записи длиной в одно предложение, заголовок. Если вместо предложения у вас получится целый абзац, шаблон поплынет.



Конечно, вы можете переинициализировать эту функцию для создания развернутых черновиков. Однако сделать это можно будет только путем глобального изменения настроек. Внесите в файл .emacs следующую строчку:

```
(setq howm-template "= %title\n%date %file\n\n%cursor")
```

Теперь во всех новых заметках будет следующее:

По умолчанию:



После настройки:



Раньше вы начинали с заголовка, а затем перемещали курсор в поле заметки. Теперь вы можете сразу писать текст заметки, а к заголовку вернуться позже.

При сохранении быстрой заметки вы получите текст после даты, а заголовок останется пустым.

* * *

Стоит отметить еще одну особенность шаблона быстрых заметок. По умолчанию в между заголовком и датой в квадратных скобках добавляется пустая строка. Это не баг, а наследие. Пустая строка играет какую-то роль в неведомых мне пакетах. Вы можете смело перенастроить эту функцию так, чтобы шаблоны обычных и быстрых заметок были одинаковыми⁶¹.

⁶¹ Подробности в разделе «Дополнительные настройки», стр. 78-79

ДУБЛИРОВАНИЕ

Для некоторых видов заметок полезно иметь заготовку. Это помогает избежать бесчисленных повторов одних и тех же действий. Например, в персонологической карточке нам потребуются графы: годы жизни, основные труды и т.п.

В howm эту задачу можно решить благодаря функции дублирования. Команда С-с , D создает клон текущей заметки. Дубликат ничем не отличается от оригинала, кроме нового нумерического имени и того, что первая строка пустая (она оставлена для нового заголовка)

Создание дубликата можно вызвать и из основного меню howm - D[Dup]. В этом случае будет продублирован последний открытый буфер.

В заметках с несколькими заголовками (см. стр. 52) дублируется только тот раздел текста, в котором находится курсор.

Это на самом деле открывает интересные возможности.

Во-первых, вы можете создать один документ с коллекцией шаблонов, разделенных заголовками.

Во-вторых, вы можете легко дублировать часть какой-либо заметки, просто добавив знак = в начале и конце избранного фрагмента.



Документ с шаблоном можно озаглавить специальным шифром, типа - Ш_01 (так их можно будет быстро вызвать через поиск).

Если вы передумали делать дубликат уже после того, как нажали команду, ничего страшного. Нажмите отмену С-/ и закройте окно. Никаких следов в системе не останется⁶².

⁶² Тут все также, как и отменой при создании заметки (см. стр. 18)

ПРОИЗВОЛЬНЫЕ НАЗВАНИЯ

Я уже говорил о том, что вы можете перенести в папку `howm` любые `plain text` файлы (`.txt`, `.md`, `.org`, `.tex`). И называться они могут как угодно. Разумеется, для нормальной работы системы лучше соблюдать последовательность. Нумерические имена хороши своей универсальностью и абстрактностью. Тем не менее файл с именем `Вася_Пупкин.txt` приживется в системе без проблем.

Кстати, в `howm` тоже можно создать файл с произвольным именем. Для этого зарезервирована команда `C-c , I`. Она открывает специальный диалог.

```
-:%%- *howmM:0000-00-00-000000.txt* Bot L26 (HM howm AL)
Memo file: ~/howm/
```

Введите желаемое имя файла и расширение. Например, `путевые_заметки.org`

Если вы хотите, чтобы неформатные заметки располагались в отдельной папке, нужно ее предварительно создать (у нее тоже может быть любое имя). Для этого лучше всего использовать штатный функционал Emacs.

`M-x make-directory RET ~/howm/имя RET`

После того, как директория создана, вы вольны разместить там любые файлы.



Но `howm` воспринимает все файлы, размещенные в его директории, как часть картотеки вне зависимости от того, в какой суб-директории они находятся. Во всех списках они будут представлены вместе.

Еще раз отмечу, что для отказа от стандартной системы заметок `howm` нужно иметь серьезный резон.

К слову, вы можете посмотреть любой текстовый файл за пределами директории `howm`, воспользовавшись командой: `C-c , M`

Если вы хотите, чтобы при этом еще и синтаксис `howm` функционировал, наберите:

`M-x howm-show-buffer-as-howm`

СИСТЕМА ПЛАНИРОВАНИЯ В HOWM

Когда речь заходит о персональном управлении знаниями, сразу понятно, что эта деятельность предполагает хотя бы минимальное планирование. Какие-то книги нужно прочесть к определенному сроку, что-то внести в картотеку, где-то переработать материал, перечитать статью и т.п.

Если картотека становится инструментом ежедневного труда, вполне разумно вести всю эту бухгалтерию прямо там. Howm предполагает систему гибкого планирования, призванного помочь неорганизованным людям. Идея в том, чтобы вместо бесплодных попыток полного искоренения хаоса, обратить его себе на пользу.

Ключевой элемент работы с информацией в howm - это списки. В планировании они более чем уместны. Однако любые списки дел имеют тенденцию разрастаться до несовместимых с продолжительностью жизни размеров. Нужно сделать так, чтобы на виду были не все задачи сразу.

Принцип здесь тот же, что и при организации заметок. Тратя времени на классификацию оказывается контрпродуктивной. Хираока пишет, что сделал решительный шаг и упразднил «важность», вместо этого он реализовал список на основе «сезонного порядка». Под этой метафорой имеется в виду доступность напоминаний в подходящее время.

В вашем распоряжении календарные события (schedule), крайние сроки (deadline), задачи (Todo), памятки (reminder) и откладывалки (defer).

Все задачи, события и т.д. прикреплены к определенным датам, но смысл этих дат отличается.

Дата в задаче обозначает время начала действий.

Дата в дедлайне обозначает время окончания действия.

Дата события обозначает конкретный день, когда что-то должно произойти.

Что насчет памяток и откладывалок?

Памятки позволяют какое-то время удерживать на виду нужную информацию. После этого включается процесс «забывания» – памятка постепенно опускается вниз списка.

Откладывалки – это особый тип памяток, которые с определенной цикличностью будут оказываться у вас на виду.

Быстрый обзор планов возможен в главном меню `howm`. Здесь генерируется список дел, событий, памяток и т.д. `Howm` извлекает эти данные из заметок, так что вам не нужно беспокоиться, где вы оставили запись. Она не потеряется.

Список в главном меню не исчерпывающий, его глубина - 50 пунктов и расписание на семь дней. Но в этом и смысл, чтобы убрать с глаз долой задачи, для которых не сезон.

Естественно, в самом начале вашей работы с `howm` секция с планами будет пустой.

```
*howmM:0000-00-00-000000.txt* - GNU Emacs at Andreis-MacBook-Pro.local
menu%
e[Remember] c[New] D[Dup] Search(s[String] g[Regexp] o[Occur])
a[All] l[Recent] A[Around] y[Schedule] t[Todo] b[Buffers] x[Marks]
K[<Title>] [Name] d[Date] i[Key] r[Update] w[Random Walk] [Killall]
. [Today] :[Yesterday] h[History] R[Update Menu] [Edit Menu] [Preference]

[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)
!deadline, +todo, -reminder, ~defer (top 50 entries)
-----
Recent
```

Если у вас намечено небольшое количество задач, они все попадут на главную страницу, независимо от релевантности. Однако когда ваш список вырастет, только приоритетные задачи останутся на виду.

Список задач, дедлайнов и т.п. в `howm` не статичен, он динамически меняется в зависимости от даты.

Хираока использует образ потопления задач (перемещения вниз списка) и всплытия задач на поверхность (перемещения вверх списка). Это позволяет увидеть актуальные на данный момент дела, так как они всегда будут занимать первые строки. Ненужное тонет, нужное остается на плаву.

Давайте разбираться, как вся эта «гидромеханика» работает.

Приоритетность – «плавучесть» – каждого элемента вычисляется на основании его свойств и текущей даты.

В `howm` существуют три различимых группы приоритетности. Элементы с приоритетностью около 77777 будут наверху списка, а элементы с приоритетностью около -77777 на самом дне. Соответственно задачи, с приоритетностью 0, – «плавают» посередине⁶³.

⁶³ По умолчанию эти цифровые значения не видны. Если вы хотите, чтобы они отображались рядом с задачей, внесите корректировку в `.emacs` (см. стр. 85 в «Дополнительных настройках»).

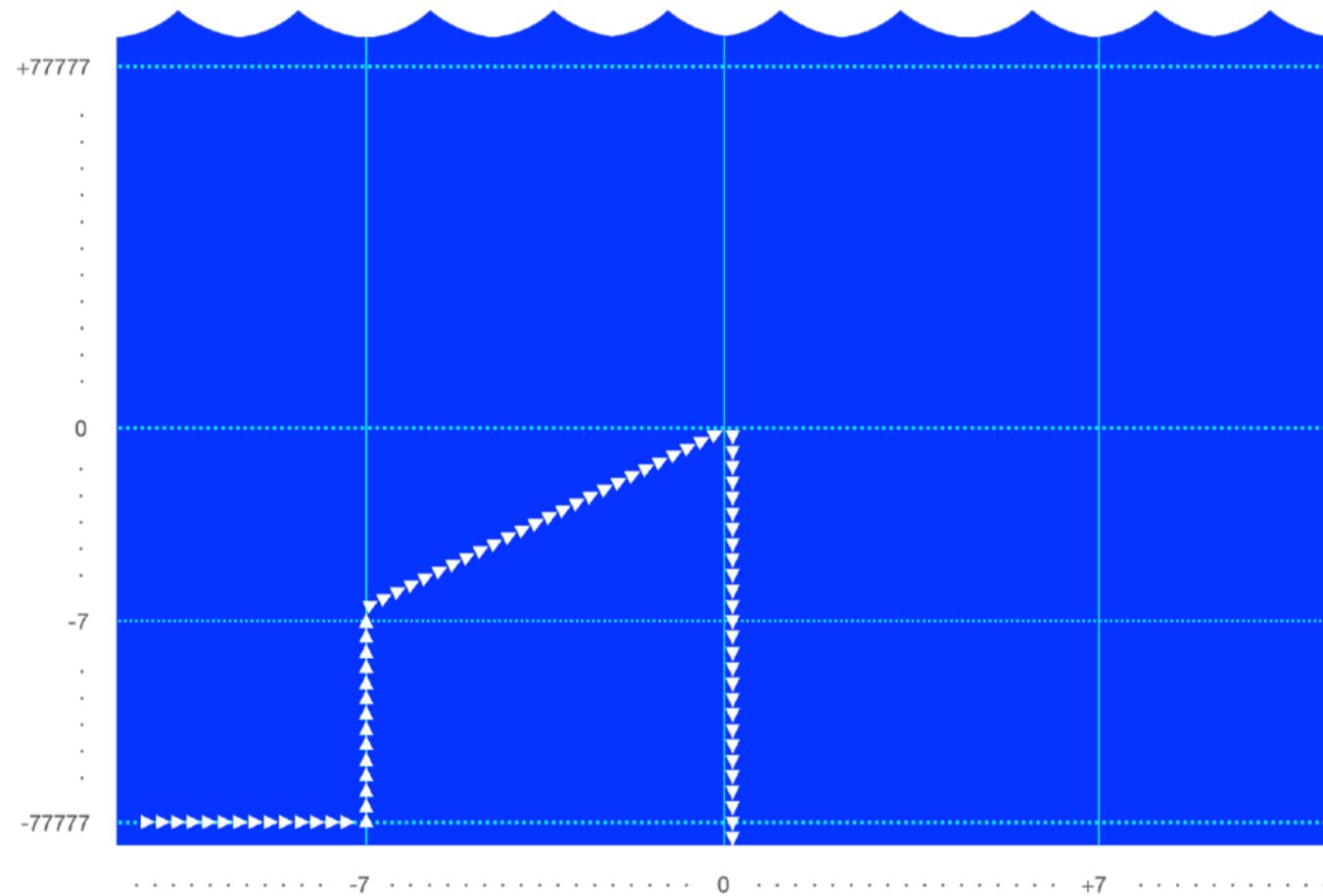
Алгоритм ранжирования задач таков, что все грядущие планы оказываются в глубине списка. Очередность этих «утопленников» не несет никакого смысла. Их нужно просто убрать с глаз долой (конечно, это работает, когда ваш список уже достаточно большой).

По мере приближения назначеннной даты дедлайны, события, памятки, откладывалки и задачи начинают вести себя по-разному.

События

Запланированное событие становится видимым в списке за семь дней до установленной даты⁶⁴. Просроченные события убираются из списка сразу на следующий день. Вполне разумно, так как вам не нужно напоминание о концерте, который состоялся вчера.

Это поведение можно проиллюстрировать следующей диаграммой:



Для обозначения разного типа планов в `howt` существует особый синтаксис, с мнемотехническими подсказками.

Чтобы запланировать событие, сделайте следующее:

1. Откройте нужную заметку,

⁶⁴ Заметьте, что событиям в главном меню отведена специальная секция (об этом в следующей главе). Это значит, что они не соревнуются с задачами и напоминалками за место в списке.

2. Укажите дату в квадратных скобках: [2024-04-08 19:15]

3. Добавьте после закрывающей скобки значок @ и описание события:

[2024-04-08 19:15]@ Съездить в Кливленд, чтобы увидеть полное затмение

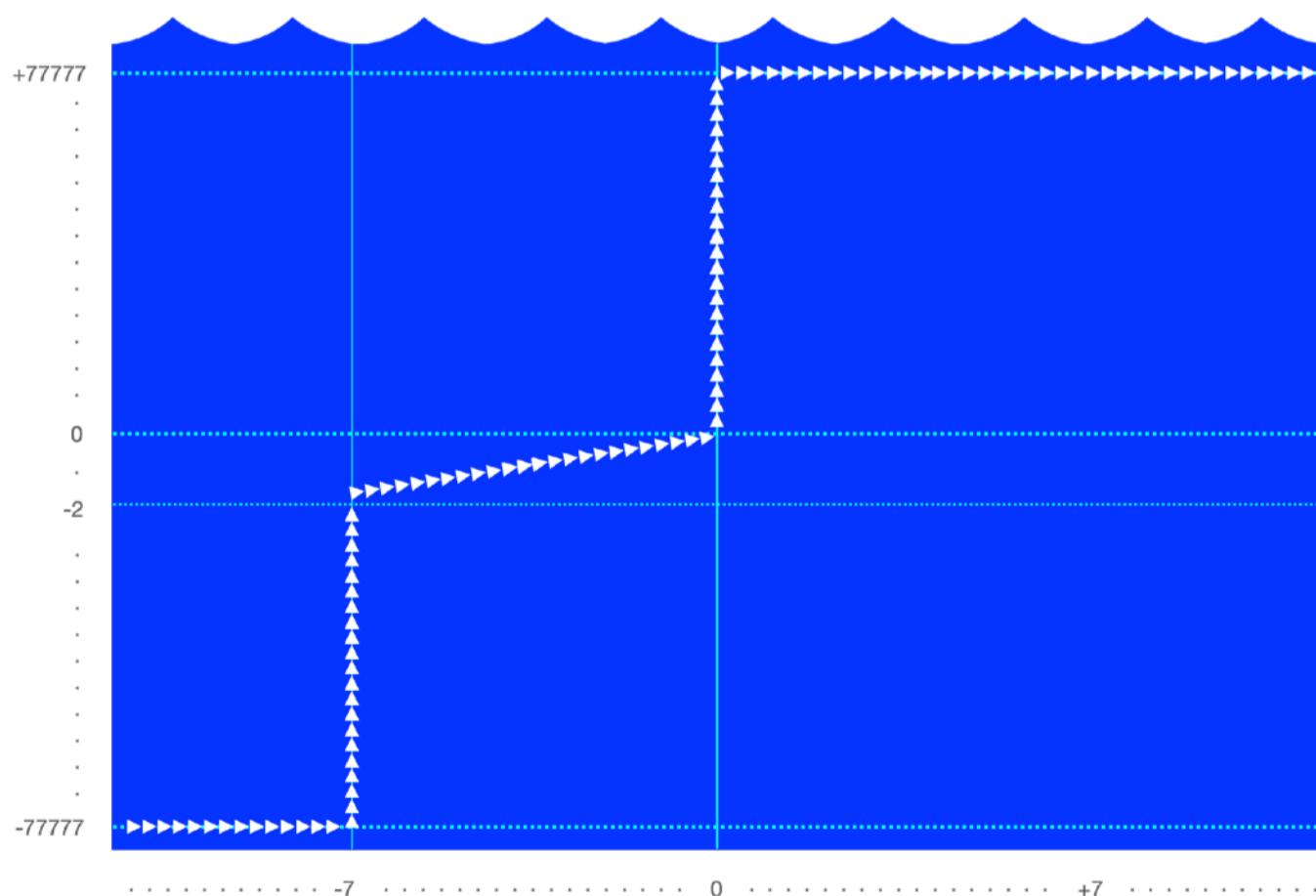
Поскольку в разной сетевой активности символ @ используется для указания адреса или человека, вам будет легко запомнить, что в howt он используется для маркировки запланированных событий (ведь они связаны с людьми и определенным местом).

Как только вы добавите @ после скобок, цвет даты поменяется на зеленый. Эта визуальная подсказка позволяет сразу разглядеть события в списках или заметках.

Дедлайны

Дедлайны находятся внизу списка, пока до назначенного срока не останется семь дней. Затем они всплывают на середину списка, где их уже можно заметить. В день X дедлайны мгновенно поднимаются на самый верх списка и остаются там даже после истечения срока.

Следующая диаграмма иллюстрирует это поведение:



Чтобы установить дедлайн, разместите восклицательный знак после квадратных скобок:

[2024-12-31]! Купить подарки семье.

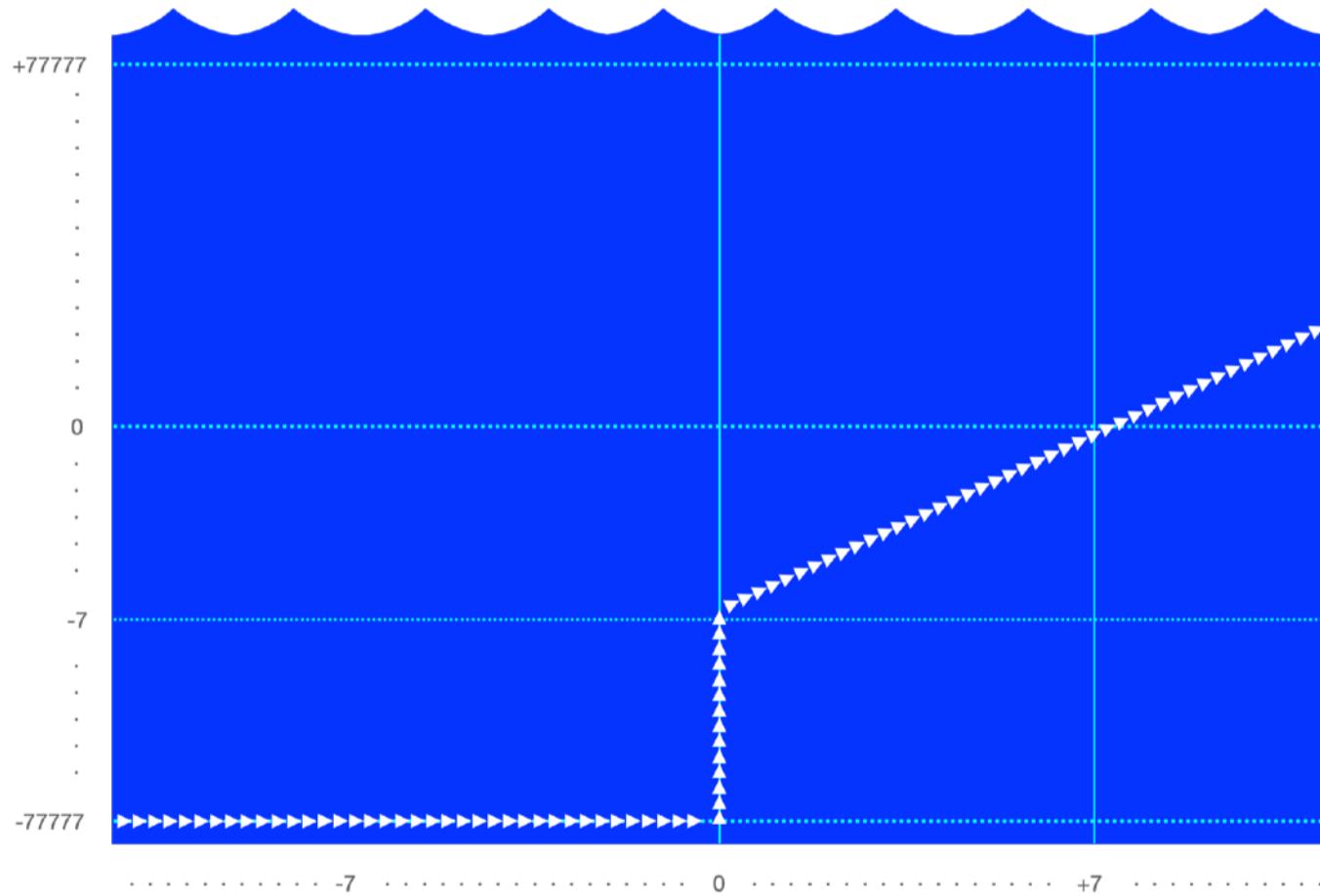
Восклицательный знак недвусмысленно указывает на важность и срочность запланированного.

Цвет шрифта автоматически изменится на красный, как только вы добавите «!» после квадратных скобок. Когда срок дедлайна истечет, восклицательный знак получит дополнительное красное выделение: **[2024–12–31]!**

Задачи

Задачи остаются внизу списка вплоть до назначенной даты. В день X они всплывают со дна ближе к середине списка, чтобы стать заметнее. В последующие семь дней задачи постепенно продвинутся в середину списка, а затем будут подниматься выше и выше. Чем старше задача, тем выше ее место в списке.

Вот диаграмма:



Чтобы назначить задачу, поместите + после квадратных скобок:

[2024–11–31]+ Позвонить гражданину N и спросить об Y

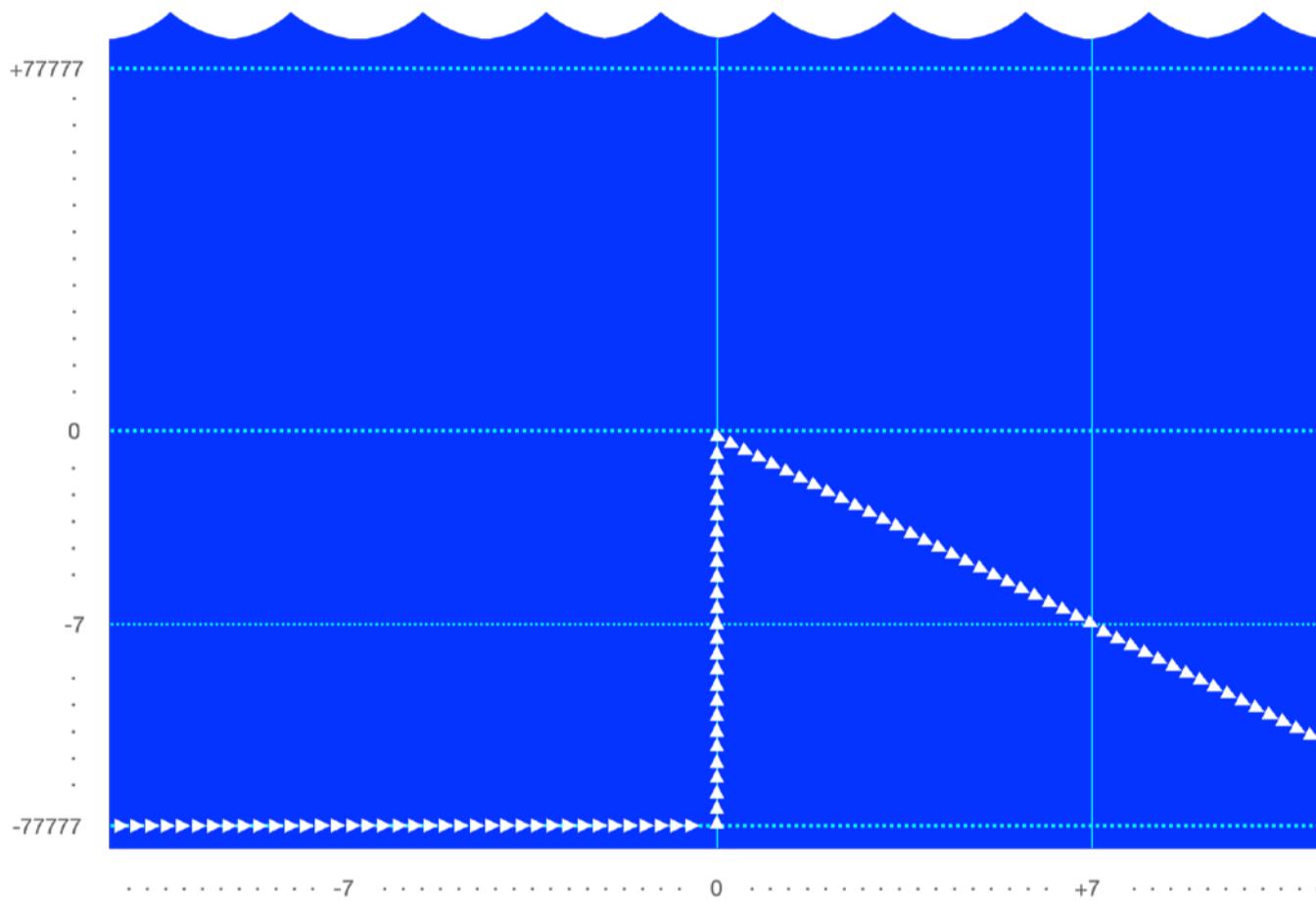
Плюс означает, что со временем важность задачи возрастает и она продвигается вверх.

Цвет шрифта, маркирующий задачи фиолетовый.

Памятки

Памятки остаются на дне списка до назначенной даты. В день X они появляются в середине списка (ниже любых просроченных дедлайнов и задач, которым больше 7 дней). После дня X памятки медленно теряют свои пункты, каждый день опускаясь все ниже в списке, пока не достигнут дна. Скорость потопления по умолчанию – один пункт в день.

Диаграмма, иллюстрирующая это поведение:



Чтобы установить памятку, разместите знак минус после квадратных скобок:

[\[2024-11-31\]– Посмотреть статью Y в журнале X](#)

Минус символизирует постепенное опускание на дно. Цвет шрифта, маркирующий памятки – синий.

Я уже упомянул, что здесь используется функция своеобразного «забывания». Так для чего же стоит использовать памятки?

Хираока объяснил их назначение так:

«По-правде я никогда не делаю памятки, привязанные к будущим событиям. Я просто использую их с текущей датой для разных вещей, которые я надеюсь сделать/изучить/прочесть/... и позволяю большинству из них утонуть, так как, разумеется, их количество превосходит мои возможности».

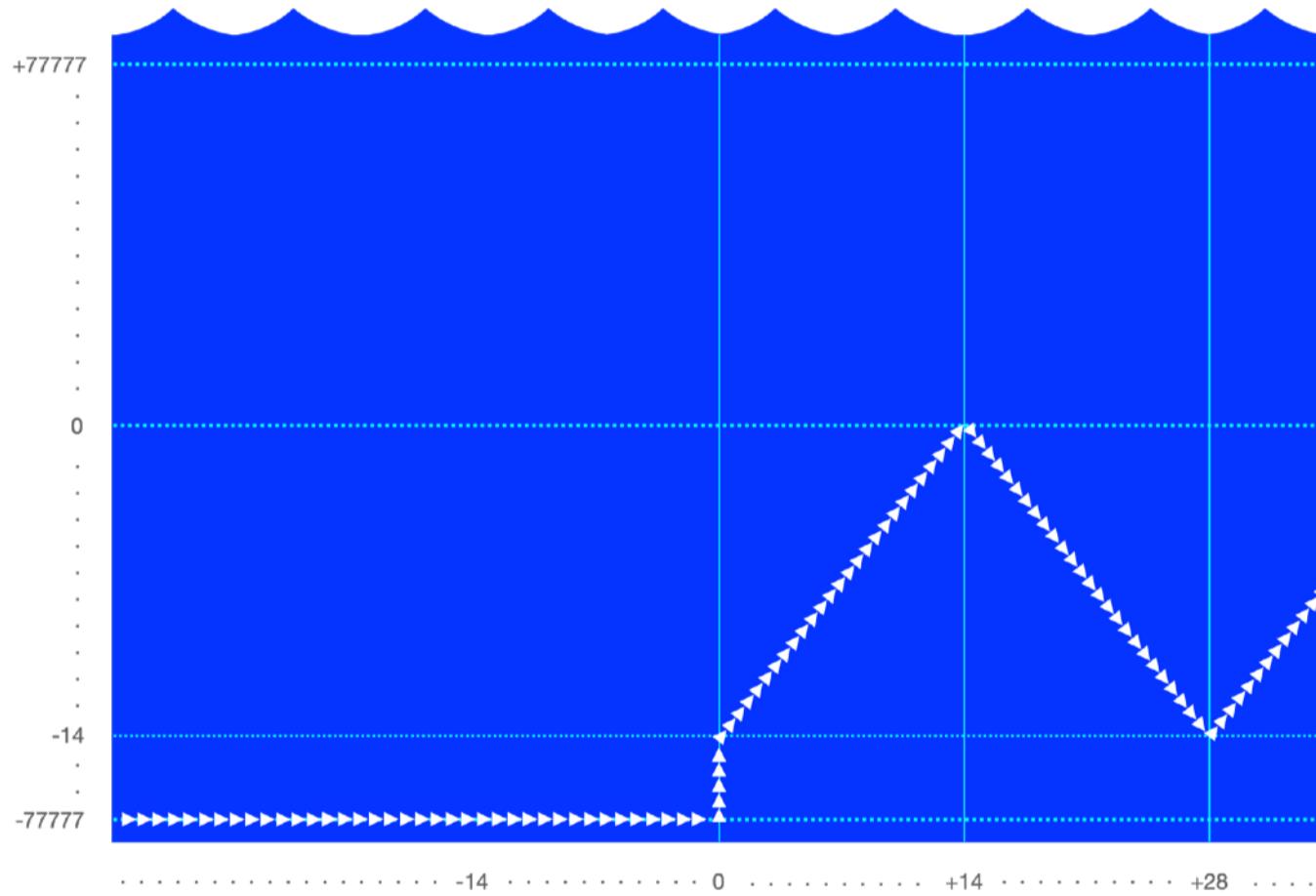


Не усложняйте процесс. Идея `howt` в минимизации стресса и давления. Она противоположна современному идеалу суперорганизованного достигателя. Дизайн `howt` намерено гибкий и снисходительный к несовершенным людям.

Откладывалки

Откладывалка поднимается ближе к срединной зоне в назначенный день (на 14 пунктов ниже нуля). Дальше за 15 дней она поднимается к середине и опять постепенно тонет на глубину -14. Этот процесс потопления и всплытия цикличен и равен 30 дням.

На диаграмме это можно отобразить следующим образом:



Символом откладывалки избрана тильда, в напоминание о волнообразных колебаниях. Для создания откладывалки разметите тильду после квадратных скобок:

[2024-06-31]~ Прочесть «Войну и мир»

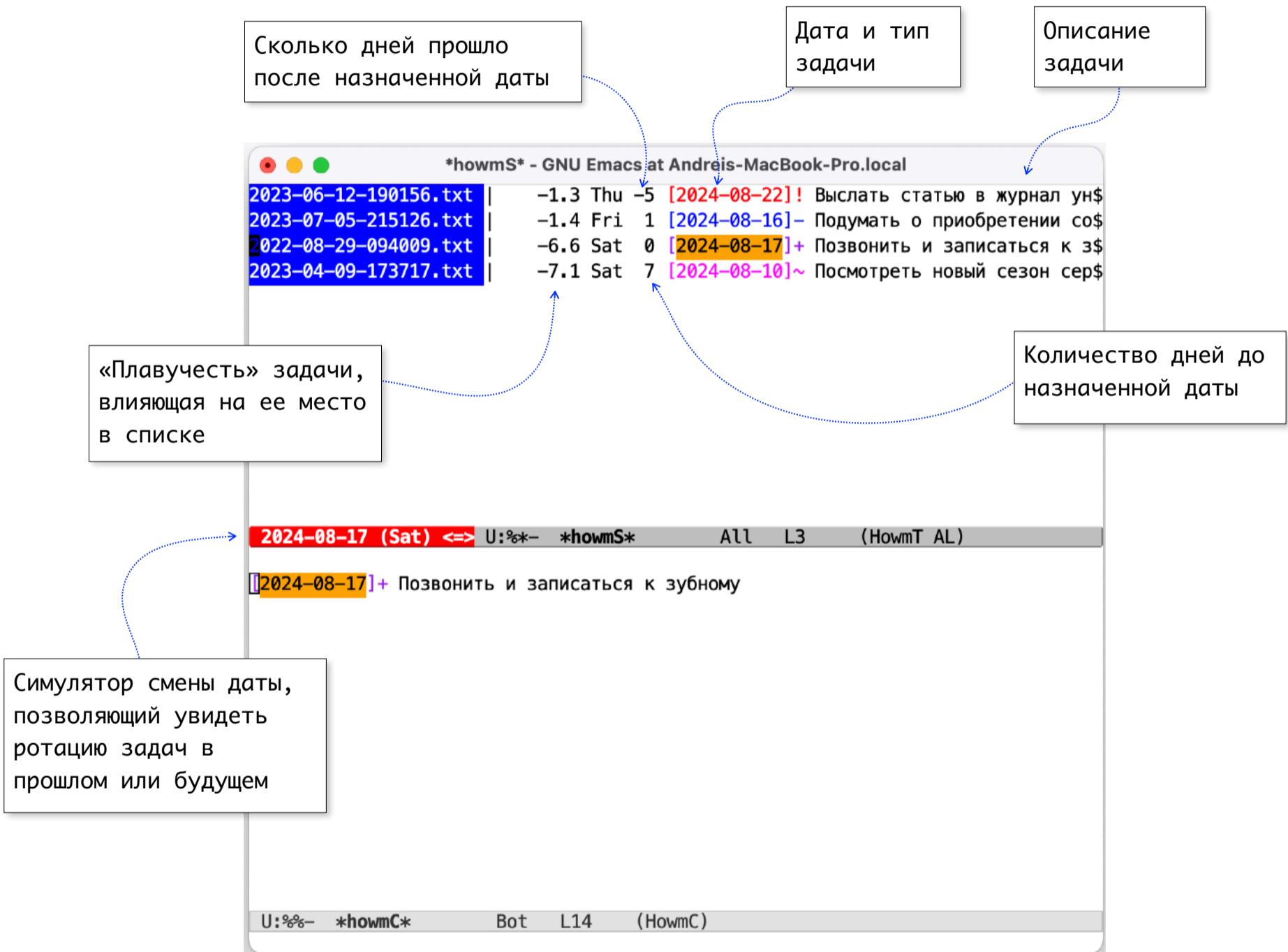
Цвет шрифта для откладывалки – розовый.

Для повторяющихся задач откладывалка не подойдет (в `howt` существует другая функция, которую мы обсудим в следующей главе). Проще всего рассматривать откладывалки как циклические памятки.

Принцип приоритизации планов в howm может показаться слишком сложным и абстрактным. К счастью, все это можно визуализировать. В howm есть специальная функция, позволяющая увидеть, как приоритизация работает.

Наберите команду: M-x howm-simulate-todo

Это запустит специальный симулятор, в котором вы сможете переключать даты и наблюдать за динамикой.



Если вы хотите увидеть, как будет выглядеть список завтра или через несколько дней нажмите клавишу **>** необходимое количество раз⁶⁵. Для путешествия в какой-то момент прошлого, воспользуйтесь клавишой **<**. Вернуться к текущему положению дел можно, нажав знак «равно».

Обратите внимание, как изменение даты влияет на приоритетность задач и их расположение в списке.

Чтобы выйти из симулятора, нажмите клавишу **q**

⁶⁵ Чтобы перепрыгнуть сразу на 30 дней вперед, используйте команду **C-u 30 >**

УПРАВЛЕНИЕ ЗАДАЧАМИ

В предыдущей главе мы рассмотрели ключевые принципы формирования списков дел в howm. Теперь пришло время поговорить, как все это использовать в таймменеджменте.

Вы уже могли заметить, что каждая задача в howm начинается с даты в формате [YYYY-MM-DD]. Конечно, вы можете и вручную набрать эти цифры, но у howm есть для этого специальная функция.

Команда **C-c , d** вставит нужную дату в открытой заметке⁶⁶. Для этого howm откроет диалог в минибуфере:

```
U:**- 2022-08-30-190438.txt Bot L30 (Text Fly/-- howm AL Wrap)
[Thu] RET(ok), +num(shift), yummdd(set), ~yummdd(repeat), .(today): █
```

Тут много всего, но сейчас мы рассмотрим самое простое действие. Чтобы вставить сегодняшнюю дату, просто нажмите точку или RET.

Что если у меня планы на 5 января 2026 года? Достаточно набрать 260105 и нажать RET.

В большинстве случаев вам не нужно указывать полную дату. Howm может сообразить, о чем идет речь.

Цифра от 1 до 31 будет истолкована как число текущего месяца⁶⁷. Цифра 803 как 3 августа текущего года.

Набрав +14 и нажав RET, вы прибавите к сегодняшнему числу четырнадцать дней.

Команда **C-c , T** вставит сегодняшнюю дату и время без дополнительных расспросов в формате: [уууу-мм-дд НН:ММ]

Если вы напечатаете в какой-нибудь заметке вот такой символ: { } и, затем, разместив курсор внутри, нажмете RET, эффект будет идентичным предыдущему⁶⁸.

⁶⁶ В меню howm есть клавиша **d[Date>]**, которая вставляет дату в последнюю открытую заметку (последний буфер). Однако, имейте в виду, что дата появится там, где располагался курсор, то есть, возможно, посередине фразы. Если вы еще открыли никаких заметок, команда не сработает.

⁶⁷ Если на дворе июнь, а вы набрали цифру 31, howm истолкует это как 30+1 и решит, что речь про первое июля.

⁶⁸ Это один из примеров функции, которая называется *action-lock*. С ее помощью в howm можно автоматизировать множество процессов (см. «Дополнительные настройки», стр. 82-83)

Так или иначе теперь вы мастер точных датировок. Что дальше? Укажите после закрывающей квадратной скобки тип задачи и ее описание. Например: [2023-12-30]! Заполнить годовой отчет.

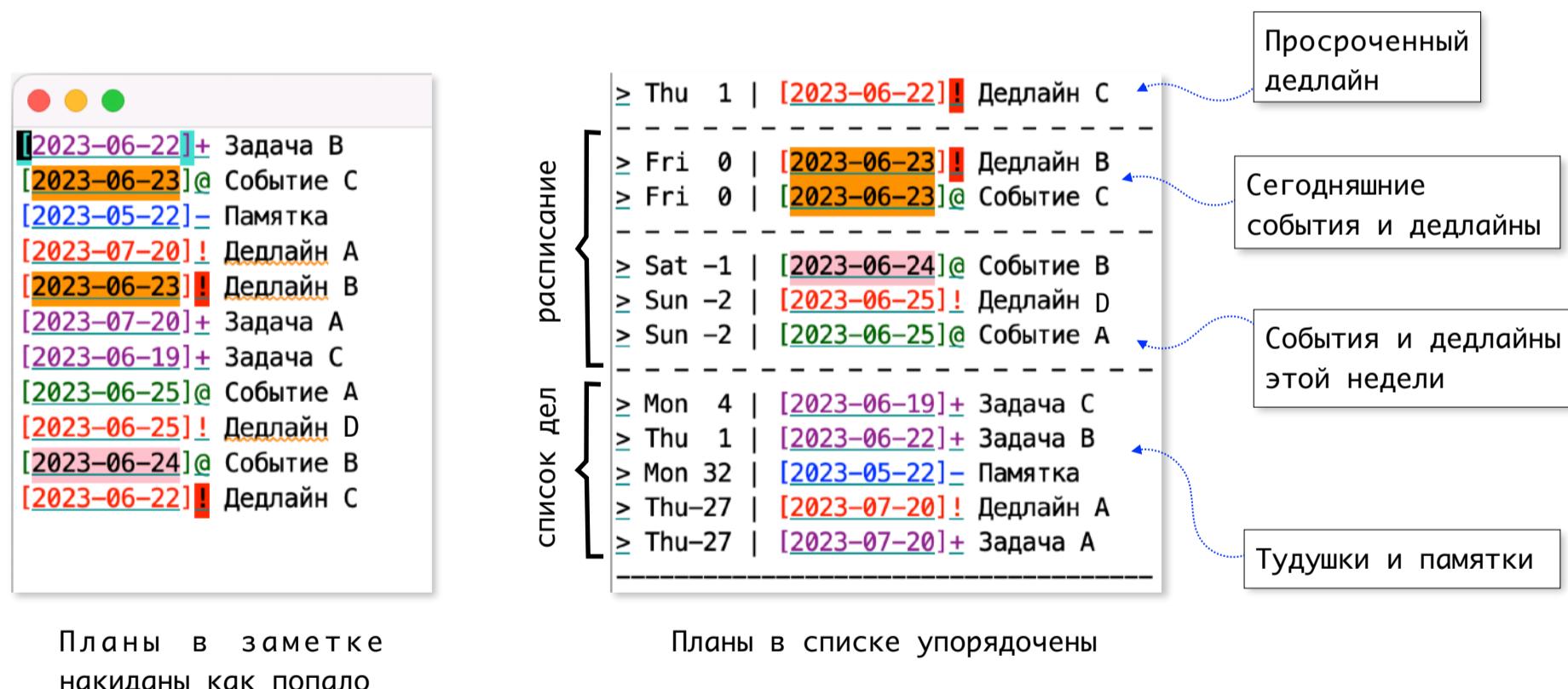
Вы можете создать отдельную заметку с планами или размещать их в разных местах. В одной заметке вы можете перечислять задачи подряд или перемежать с комментариями. Это не имеет значения. Howm позволяет быстро ориентироваться в любом многообразии задач.

Существует два способа обзора задач: через основное меню howm или через вызов специального списка. Рассмотрим обе опции.

Меню howm

Раздел с задачами в меню howm уже несколько раз упоминался (см. стр. 13 и 63). Это очень удобный способ оставаться в курсе текущих задач и событий. Неважно как разбросаны ваши планы по заметкам, howm скомпонует их и отобразит в меню.

Задачи, памятки и откладывалки в списке основного меню будут вместе. События будут выделены в отдельный блок расписания. Дедлайны в зависимости от срока могут находиться или в списке с задачами или в блоке расписания⁶⁹.



С левой стороны от назначенной даты будет указана дополнительная полезная информация. Вы можете проверить на какой день недели выпадает событие, сколько дней осталось до момента X или прошло после него.

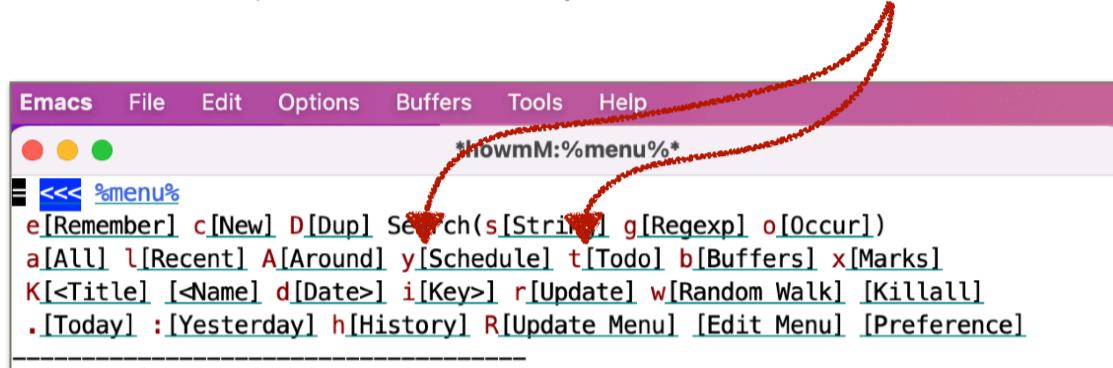
⁶⁹ Расписание показывает события и дедлайны в порядке намеченных дат. С другой стороны, памятки и задачи отображаются в «сезонном порядке».

Обратите внимание на две цветовые подсказки, помогающие анализировать планы. Сегодняшняя дата подсвеченна оранжевым, а завтрашняя – розовым⁷⁰.

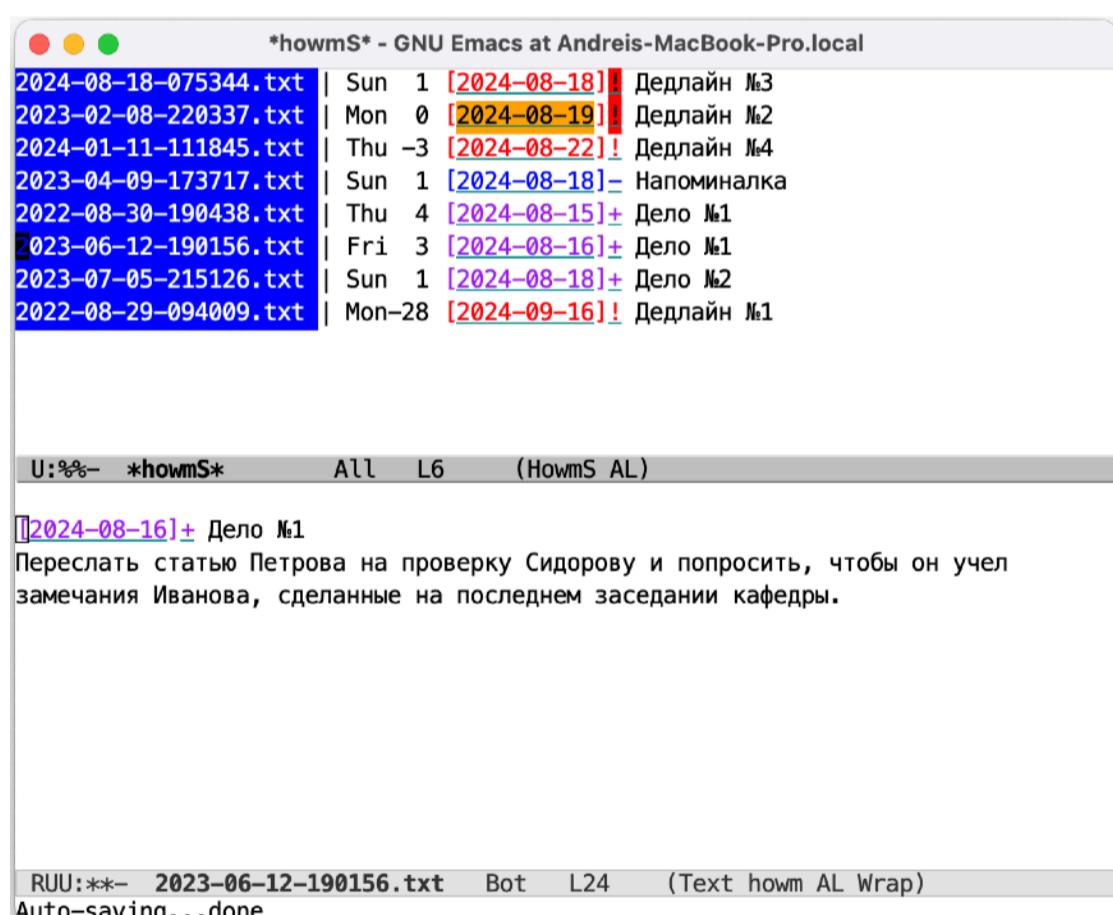
Разместив курсор на значке > и нажав RET вы перейдете в соответствующую заметку.

Списки с планами

Вы можете просмотреть список дел и событий не только в основном меню. В Howm есть две специальные команды, позволяющие работать с планами.



Список дел можно вызвать из любого места командой С-с , т. Вы получите интерфейс, привычный по работе с заметками. Преимущество (особенно, если задачи рассредоточены по разным файлам) в том, что вы можете осуществить быстрый предпросмотр контекста каждой задачи.



Список запланированных событий вызывается командой – С-с , у

⁷⁰ Подсветка будет работать не только в основном меню, но и в заметках и списках.

Дедлайны будут включены и в список дел, и в список событий. Их невозможно не заметить...

В отличие от основного меню `howt` списки не лимитированы количеством пунктов. Вы получаете доступ ко всем планам⁷¹.

Конечно, если планов много, легко растеряться. Нажмите на точку, и курсор переместится к делам или событиям, назначенным на сегодня.

Списки дел и событий ранжированы в порядке «сезонности». Если вы хотите увидеть календарную последовательность, воспользуйтесь функцией сортировки. Для этого в открытом списке нажмите `S` и выберите опцию `reminder`. `Howt` перестроит список в согласии с датами.



Вы можете интегрировать планы, размещенные в `howt`, с пакетами `Emacs` для календаря. Прежде всего, стоит обратить внимание на пакет `emacs-calfw` (<https://github.com/kiwanami/emacs-calfw>).

Хорошая система планирования должна быть гибкой. Все течет, как говорил Гераклит. Меняются обстоятельства, меняемся мы... У нас должно быть средство быстрой корректировки записей.

Чтобы перенести дату, разместите курсор внутри квадратных скобок и нажмите `RET`. `Howt` откроет диалог, который мы уже рассматривали на стр. 24.

Для того, чтобы просто перенести дело, скажем, на четыре дня, наберите `+4` и нажмите `RET`. Можно наоборот переписать задачу на сегодня, нажав точку, или указать число от 1 до 31, которое будет интерпретировано как дата (месяц при этом изменен не будет). Если вы хотите изменить и число и месяц, укажите дату в формате `MDD`, Например, `901` для первого сентября.

Переназначение даты можно осуществить не только, находясь в заметке, но и из основного меню `howt`. Действия те же. Переместите курсор внутрь квадратных скобок с датой и нажмите `RET`. `Howt` временно перебросит вас в нужный файл с открытым диалогом в минибуфере. После выполнения операции вы автоматически вернетесь в меню⁷².

Иногда нужно изменить не дату, а тип мероприятия. Памятка преобразуется в дедлайн, событие в памятку и т.п.

⁷¹ Для примера, события, назначенные на следующий квартал, из основного меню видны не будут.

⁷² Учтите, что изменения пока не сохранены. Для того, чтобы сделать это, не покидая меню, используйте команду `C-x S`

Разместите курсор на символе после квадратных скобок и нажмите RET. В минибуфере появится диалог:

U:--- 2022-08-30-190438.txt Bot L41 (Text Fly/— howm AL Wrap)
RET (done), x (cancel), symbol (type), num (laziness): █

Введите символ (~ @ - ! +), соответствующий новому статусу, и нажмите RET.

Так же как и в случае с датами, вы можете вносить изменения в записи прямо из главного меню howm⁷³.

Кроме изменения типа мероприятия вы можете задать параметры его поведения (быстроны всплытия/потопления в списке). Для этого после символа (~@-!+) нужно указать цифру.

В минибуфере, на иллюстрации выше, есть опция: num (laziness). Вам достаточно указать нужную цифру и нажать RET.

Разберем на примерах.

Вы можете удерживать запланированные события в меню howm даже по прошествии намеченной даты. Например, указав [2023-06-25]@5 вы сделаете так, что событие будет на плаву еще пять дней.

Если вы хотите, чтобы предупреждение о дедлайне начало всплывать раньше, чем за семь дней до назначеннной даты, укажите требуемое число. Например: [2023-01-20]!14

Скорость затопления памятки может быть больше. Запись [2023-08-15]-7 будет терять один пункт в приоритетности каждые семь дней.

Чтобы замедлить всплытие задачи, укажите число больше семи (цифра назначена по умолчанию). Например, задача [2023-06-25]+14 будет всплывать в два раза медленнее обычной.

В случае с откладывалкой число будет означать длину цикла потопления-выныривания. Чтобы удвоить частоту колебаний, укажите: [2023-03-09]~15

Если вы не можете вспомнить стандартные значения всплытия и погружения для разных элементов, обратитесь к подсказке в меню howm (см. стр. 13)

Подошел черед поговорить о смене статуса планов. Рано или поздно настанет момент, чтобы выполнить или бросить задуманное.

⁷³ Конечно, если вы редактируете планы, находясь в самой в заметке, достаточно просто стереть символ и впечатать новый (количество кликов одинаковое).

Расположите курсор на символе после квадратной скобки и нажмите RET. В появившемся диалоге минибуфера (см. предыдущую страницу) есть две опции.

Если вы еще раз нажмете RET, дело будет считаться завершенным. Его отображение изменится⁷⁴.

[2023-06-04]+ достичь просветления



[2023-06-28]. [2023-06-04]:+ достичь просветления

Знаком завершения дела является точка. Автоматически howt добавляет дату финала. Таким образом, у вас под рукой удобный инструмент для анализа.

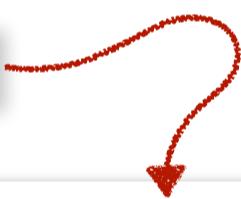
Однако вы можете использовать и упрощенный синтаксис. Разместите где-нибудь примечание: [2023-06-28]. достичь просветления

И таким образом вы будете знать, что в деле поставлена точка.

Другой возможный исход – отмена плана. За всем не угнаться, ничего страшного.

Находясь в диалоге минибуфера, нажмите клавишу X

[1990-12-31]+ Прочесть «Войну и мир»



[2023-06-28]. cancel [1990-12-31]:+ Прочесть «Войну и мир»

Примечание cancel будет напоминать, что дело не завершено, а отменено. В остальном синтаксис не отличается.

Завершенные и отмененные планы – утопленники, они всегда находятся на дне списка. В главном меню они отображаться не будут.



Просмотреть выполненные планы можно с помощью команд C-c , t или C-c , u

Поскольку завершенные дела находятся на дне, нажмите R, чтобы увидеть перевернутый список

Последняя функция, которую осталось рассмотреть в этом разделе: повторяющиеся задачи.

⁷⁴ Заметьте, что как только вы завершите задачу, внизу экрана (в минибуфере) появится сообщение: Finished 1 task! Это поздравление howt будет показывать всякий раз, подсчитывая количество завершенных за сессию дел. См. также раздел «Дополнительные настройки», стр. 84

Howm располагает только вариантом ограниченного количества повторов. У такого подхода есть свои плюсы и минусы.

Минусы очевидны. Вы не можете единожды создать напоминание о циклических задачах. Вы не сможете узнать, какое важное событие придется на второй понедельник декабря в следующем году (правда, случаи, когда необходима информация о сильно отсроченных событиях, редки).

Плюсы ограниченных повторов в том, что вы сознательно принимаете решение, сколько раз следует предпринять действие, стоит ли продолжать.

Чтобы создать повторяющуюся задачу/событие, разместите курсор внутри квадратных скобок и нажмите RET. Перед вами возникнет уже известный диалог минибуфера:

```
U:--- 2023-04-10-140557.txt All L2 (Text Fly/-- howm AL Wrap)
[Mon] RET(list), +num(shift), yummdd(set), ~yummdd(repeat), .(today):
```

Вам нужно указать дату окончания цикла. Например, ~250601

Дальше нажмите RET, и howm уточнит необходимую частоту повторов.

```
U:**- 2022-08-30-190438.txt Bot L31 (Text Fly/-- howm AL Wrap)
Every? [RET(all), num(days), w(week), m(month), y(year)]
```

RET – ежедневно до указанной даты.

num – любая цифра будет количеством дней между повторами.

w – еженедельно в тот же день, что и в начальной дате.

m – ежемесячно, того же числа, что и в начальной дате.

y – ежегодно, того же месяца и числа, что в начальной дате.

Представьте, что у вас есть задача, которую не завершить за один подход:
[2023-01-01]+ Познать самого себя

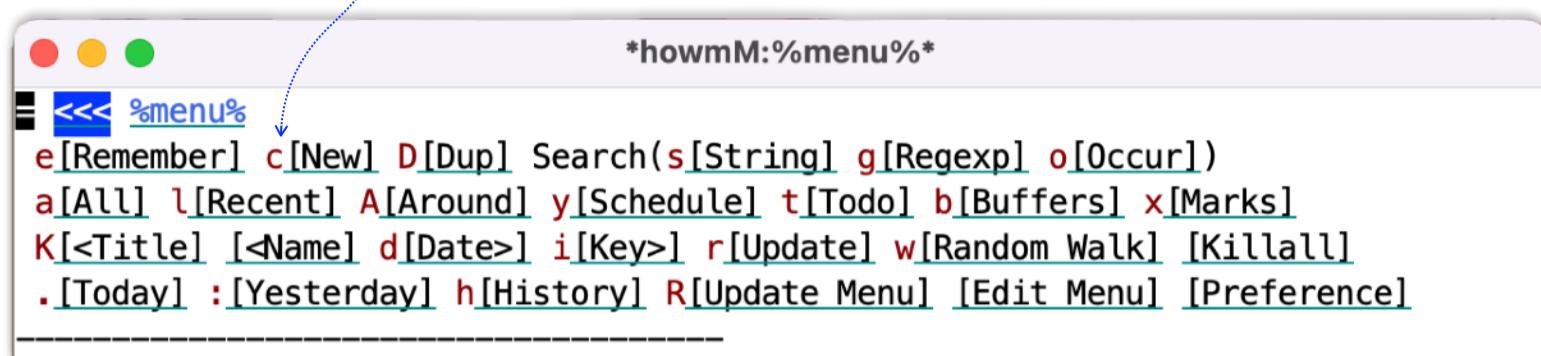
Поместите курсор на дату и нажмите RET. Введите дату финала: ~231231

Определите частоту повторов – например, каждые 42 дня – и нажмите RET.

Вуаля! Вы получили девять подходов к задаче в год. Howm клонирует задачу указанное количество раз, изменяя число. Плюс в том, что затем вы можете переназначить дату отдельного элемента, не ломая остальную цепочку (примерно также это работает в org-mode).

КОРОТКИЕ КОМАНДЫ В СПИСКАХ

Как вы знаете, находясь в меню howm, вам не нужно использовать длинные клавиатурные сочетания, типа **C-c , c**. Для выполнения команды достаточно просто нажать **C**



То же самое касается множества команд в списках howm.

Вы можете пропустить первую часть и использовать лишь последний символ.

C-c , a

Одним нажатием вы можете переключаться между разными списками и выполнять множество других действий.

a - показать список всех заметок

l - показать список недавних заметок

y - показать список запланированных событий

t - показать список задач

b - показать список открытых буферов

M - открыть файл по имени

s - открыть поиск по терминам и ключевым словам

g - открыть поиск по regexp

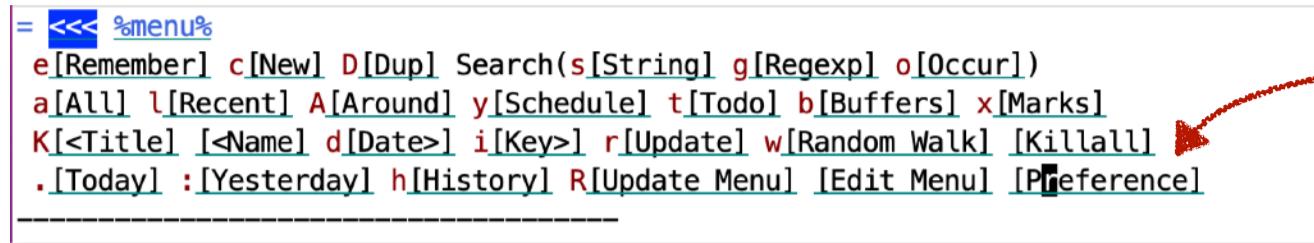
c - создать новую заметку

e - создать быструю заметку

, - открыть меню howm

ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ

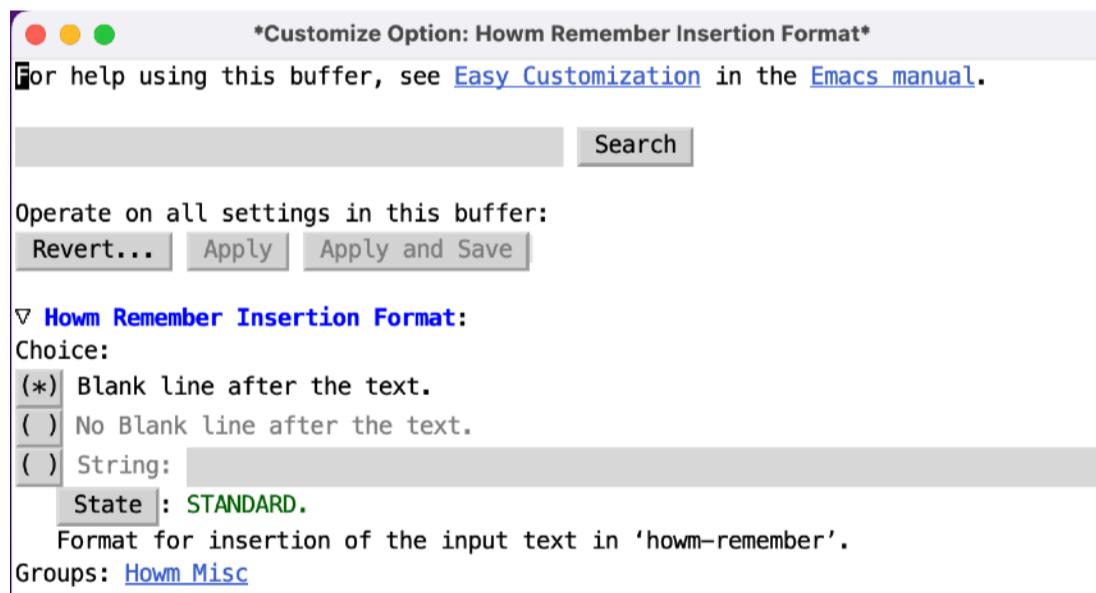
Вы можете перейти к настройкам `howm` прямо из основного меню⁷⁵.



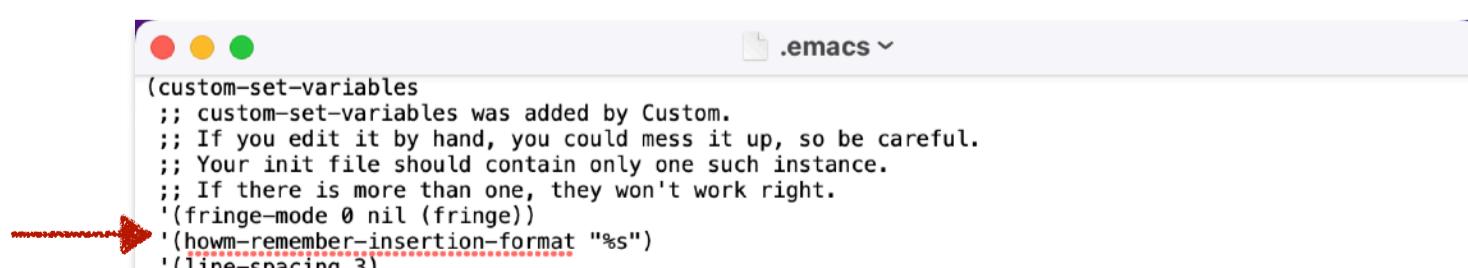
Вы можете изменить любые значения. Чтобы изменения вступили в силу, не забудьте нажать **Apply and Save**. Ваши предпочтения будут записаны в файл `.emacs` в раздел `custom-set-variables` (с них документ начинается).

Если вы передумали, изменения можно убрать, зайдя в соответствующий раздел `howm group` и нажав **State**. В открывшемся диалоге вы увидите опцию **Erase Customisation**. Нажмите на нее и затем на **Apply and Save**. Можно просто удалить соответствующие строки из `custom-set-variables`. `Howm` забудет о всех модификациях.

Рассмотрим конкретный пример. Допустим вы хотите убрать пустую строку в шаблоне быстрых заметок. Проследуйте в настройки: **Preference** → **Howm Misc** → **Howm-remember-insertion-format**. Выберите вторую опцию: No Blank Line after the text



После этого нажмите на **Apply and Save**. Теперь, заглянув в файл `.emacs`, вы обнаружите новую строчку.



⁷⁵ Альтернативный способ попасть в настройки `howm`: M-x customize → Applications → Howm.

Если вы решите изменить настройки, Emacs автоматически ее перепишет.

Опытные пользователи часто прибегают к прямой редактуре. Вы можете и сами записать в файле `.emacs` соответствующую команду:

```
(setq howm-remember-insertion-format "%s")
```

Однако следует иметь в виду, что отмотать такие изменения назад через настройки уже не получится. Нужно будет опять залезать в `.emacs` для коррекции⁷⁶.

* * *

Поскольку навык работы непосредственно с `.emacs` не будет лишним, ниже мы рассмотрим продвинутые варианты настроек. В них нет ничего особенно сложного. Это код, который указывает `howm`, что и как делать.



Для того, чтобы служебные файлы были аккуратно упакованы в папке `howm` в вашей домашней директории, запишите в `.emacs` следующее:

```
(setq howm-home-directory "~/howm/")
(setq howm-directory "~/howm/")
(setq howm-keyword-file (expand-file-name ".howm-keys" howm-home-directory))
(setq howm-history-file (expand-file-name ".howm-history" howm-home-directory))
```



Если есть желание сохранять файлы в определенном формате (в примере `markdown`) пишем:

```
(setq howm-file-name-format "%Y/%m/%Y-%m-%d-%H%M%S.md")
```



Файл с меню (`0000-00-00-000000.txt`) находится у `howm` в той же папке, что и заметки. В результате, он попадает в общий список наряду с другими записями. Если вам это мешает, можно перенести его за пределы папки `howm`. Не забудьте указать в `.emacs` новое место обитания:

```
(setq howm-menu-file "~/путь к файлу/0000-00-00-000000.txt")
```

⁷⁶ Если вы вносите одни изменения через панель настроек, а другие напрямую, возможен конфликт. Желательно вести для себя протокол изменений, чтобы не запутаться.

Целесообразность такого решения сомнительна (чем меньше папок, тем лучше). Однако возможность есть.



Вы можете указать дополнительные директории, в которых `howm` следует искать заметки:

```
(setq howm-search-path '("/path/to/notes/directory" "/path/to/other/notes/directory"))
(setq howm-search-other-dir t)
```



Если вы хотите интегрировать `howm` с работой почтового клиента под `Emacs`, пропишите в `.emacs` следующие строки (естественно, нужно указать соответствующий путь к файлам):

```
;; search ~/Mail and ~/News in addition to normal notes
(setq howm-search-path '("~/Mail" "~/News"))

;; initial value of M-x howm-toggle-search-other-dir
(setq howm-search-other-dir t)
```



Еще одна полезная настройка:

```
(setq howm-view-keep-one-window t)
```

Это позволяет отключить нижнее окно предпросмотра (клавиша 1) и листать список заметок⁷⁷. Вернуться к режиму предпросмотра можно нажав клавишу 2, пробел или v



Когда количество заметок переваливает за тысячу, `howm` становится задумчивым при загрузке меню. Можно оптимизировать процесс, прописав в `.emacs` следующие строки:

```
(setq howm-view-use-grep t) ;; use external grep
(setq howm-menu-refresh-after-save nil)
(setq howm-menu-expiry-hours 1) ;; cache menu N hours
(setq howm-menu-file "0000-00-00-000000.txt") ;; don't *search*
```

Это будет экономить вам незначительное время, но работать станет приятнее.

⁷⁷ Без этой настройки при перемещении курсора по списку `Howm` будет снова инициировать предпросмотр.

Есть побочные эффекты! Меню перестанет обновляться автоматически. Но это не страшно. В самом меню есть команда обновления содержания - R[Update Menu]. Другой побочный эффект - поиск в `howm` становится чувствительным к регистру.



Ссылка-сюда чувствительна к регистру. Если слово в `.howm-keys` написано со строчной буквы, то же слово с прописной буквы подчеркнуто в заметках не будет. Если перейти по ссылке, с списка будут все упоминания слова, вне зависимости от регистра (поиск к регистру не чувствителен).

Если вы хотите это исправить, пропишите в `.emacs` следующую строку:

```
(setq howm-keyword-case-fold-search t) ; нечувствительность к регистру
```

Обратите внимание, что если вы настроили поиск с привлечением grep, функция не будет работать корректно.



Еще одна потенциально полезная настройка – переименование буфера. При просмотре открытых буферов нумерическое имя малоинформативно. Следующие две строки, добавленные в `.emacs` решают эту проблему. Вместо нумерического имени вы будете видеть в списке заголовки заметок. Это значительно облегчит навигацию. Конечно, в реальности нумерические имена останутся нетронутыми, поменяется только отображение.

```
(add-hook 'howm-mode-hook 'howm-mode-set-buffer-name)
(add-hook 'after-save-hook 'howm-mode-set-buffer-name)
```

В длинных заметках с несколькими заголовками эта настройка начинает вести себя странно. Иногда нумерическое имя буфера заменяется не на первый заголовок, а на второй или на третий.



Если вы хотите, чтобы `howm` создавал один файл на каждый день, укажите в `.emacs` следующее:

```
(setq howm-file-name-format "%Y/%m/%Y_%m_%d.txt") ; один файл в сутки
```

В этом случае `howm` будет приписывать новые записи в уже созданную заметку, снабжая из заголовками с датой.

Работать это будет так, как описано на стр. 52-55, где речь шла о длинных заметках.



Когда вы нажимаете RET на символе напоминания (!+-~@.), Howm открывает диалог, спрашивая о следующем действии. Можно сделать так, чтобы нажатие клавиши RET означало завершение задачи. Для этого пропишите в .emacs следующую строку:

```
(setq howm-action-lock-reminder-done-default "")
```

Вызов диалога в этом случае будет выполняться командой - С-и RET



Если вы откроете документ в формате org за пределами директории howm, клавиатурные команды срабатывать перестанут. Вам необходимо специально указать Emacs, что в конкретном документе режим howm должен быть включен⁷⁸. Для этого служит команда: M-x home-mode

Если вы хотите, чтобы документы org дружили с howm по умолчанию и все команды в них срабатывали, добавьте в .emacs следующую строчку:

```
(add-hook 'org-mode-hook 'howm-mode)
```

Плюс в том, что howm будет реагировать на команды из любого документа org, вне зависимости от его расположения.

Еще больший плюс - ссылки howm будут работать в любом документе org (ссылаясь, правда, можно будет только на документы в директории howm, так как именно там осуществляется поиск).

Имейте в виду, что эти настройки повлияют на функцию переключения между буферами и закрытия буферов (см. гл. «Работа с буферами»).

Если в каком-то документе org вам потребуется отключить режим howm, воспользуйтесь той же командой, которую мы использовали для включения режима: M-x home-mode



В howm используется множество функций, которые называются action-lock. Вам уже знаком значок {_,}, превращающиеся в сегодняшнюю дату, если нажать RET внутри скобок. Вы можете настроить похожий «тумблер» для выполнения других действий.

⁷⁸ При этом документы с расширением org, размещенные в директории howm, будут функционировать в том и в другом режиме без всяких проблем.

В качестве примера в `howm` уже помещен один переключатель. Напишите `{ }` в любой заметке. Разместите курсор внутри и нажмите RET. Вместо пробела появится астериск - `{*}`. Нажмите RET еще раз и символ переключится на тире - `{-}`. Еще раз - возврат к пробелу.

Вы вольны прописать для этой функции какие угодно символы, в каком угодно количестве. Например:

```
(setq action-lock-switch-default '("{ }" "{○}" "{○}" "{●}" "{●}" "{●}"))
```

Можно использовать эти символы для обозначения степени готовности заметок или завершенности задачи.

Чтобы все работало корректно, необходимо разместить эту строку в файле `.emacs` раньше записи (`require 'howm`).



Вы можете преобразовать начальную страницу `howm` по своему вкусу.

Например, задать число недавних заметок, которые вы хотите видеть в основном меню (по умолчанию их 20).

```
(setq howm-menu-recent-num 10)
```



Вы можете изменить вид главной страницы `howm` путем прямого хирургического вмешательства. Это обычный файл `0000-00-00-000000.txt`

Вы можете открыть его в любом текстовом редакторе. В меню `howm` есть специальная кнопка, которая переводит в режим редактирования - [Edit Menu]. Нажав на нее, вы увидите следующее:

```
0000-00-00-000000.txt
<<< %menu%
%"e" [Remember] %"c" [New] %"D" [Dup] Search(%"s" [String] %"g" [Regexp]
%"o" [Occur])
%"a" [All] %"l" [Recent] %"A" [Around] %"y" [Schedule] %"t" [Todo] %"b" [Buffers]
%"x" [Marks]
%"K" [<Title] [<Name] %"d" [Date] %"i" [Key>] %"r" [Update] %"w" [Random Walk]
[Killall]
%"." [Today] %":|[Yesterday] %"h" [History] %"R" [Update Menu] [Edit Menu]
[Preference]

[Schedule, Todo] -- @schedule, !deadline (until %sdays days from now)
!deadline, +todo, -reminder, ~defer (top %tnum entries)
%reminder

Recent
%recent

Random -- If you do not like this, [Edit Menu] to erase it.
%random
```

Вы можете поменять местами любые элементы в этом тексте. Например, переместить кнопку "%h" [History] на первое место в меню команд. Или поменять местами недавние и случайные заметки. Даже вовсе убрать какой-то элемент⁷⁹.

Вы также вольны добавить дополнительные элементы в меню. Это может быть ссылка на какую-то заметку МОС, точка входа в систему.

Просто запишите где-нибудь обычную для howm ссылку:

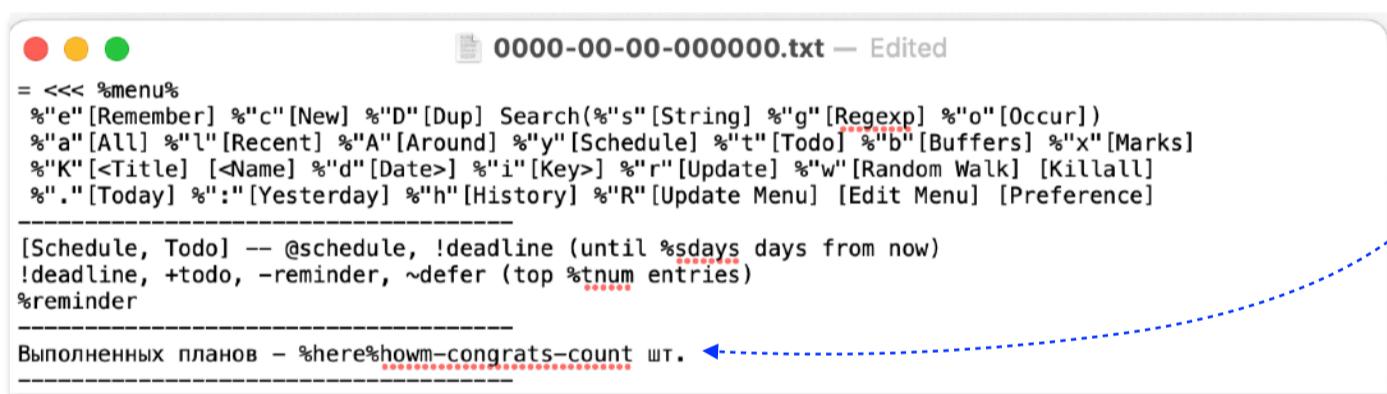
```
>>> Индекс/философия
```

Еще один вариант – добавить поисковый фильтр по определенному слову. Для этого разместите в файле начальной страницы следующую запись:

```
%here%(howm-menu-search "слово для поиска")
```

В меню это будет отображаться как список заметок.

Можно разместить в меню и подсчет выполненных планов. Для этого запишите в избранном месте следующую строку:

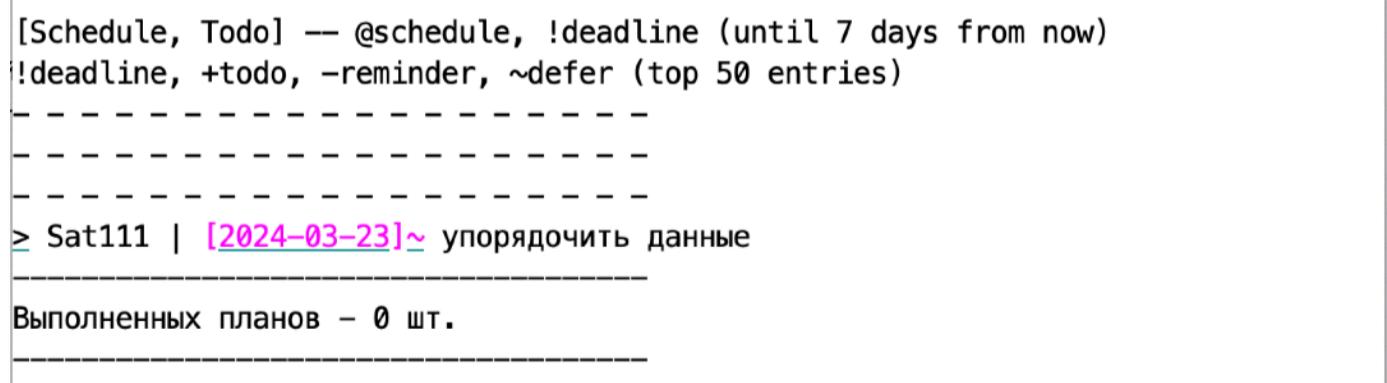


```
= <<< %menu%
%"e" [Remember] %"c" [New] %"D" [Dup] Search(%"s" [String] %"g" [Regexp] %"o" [Occur])
%"a" [All] %"l" [Recent] %"A" [Around] %"y" [Schedule] %"t" [Todo] %"b" [Buffers] %"x" [Marks]
%"K" [<Title] [<Name] %"d" [Date] %"i" [Key] %"r" [Update] %"w" [Random Walk] [Killall]
%"." [Today] %"::" [Yesterday] %"h" [History] %"R" [Update Menu] [Edit Menu] [Preference]

[Schedule, Todo] -- @schedule, !deadline (until %sdays days from now)
!deadline, +todo, -reminder, ~defer (top %tnum entries)
%reminder

Выполненных планов - %here%howm-congrats-count шт.
```

Теперь при вызове меню вы всегда сможете убедиться в собственной продуктивности.

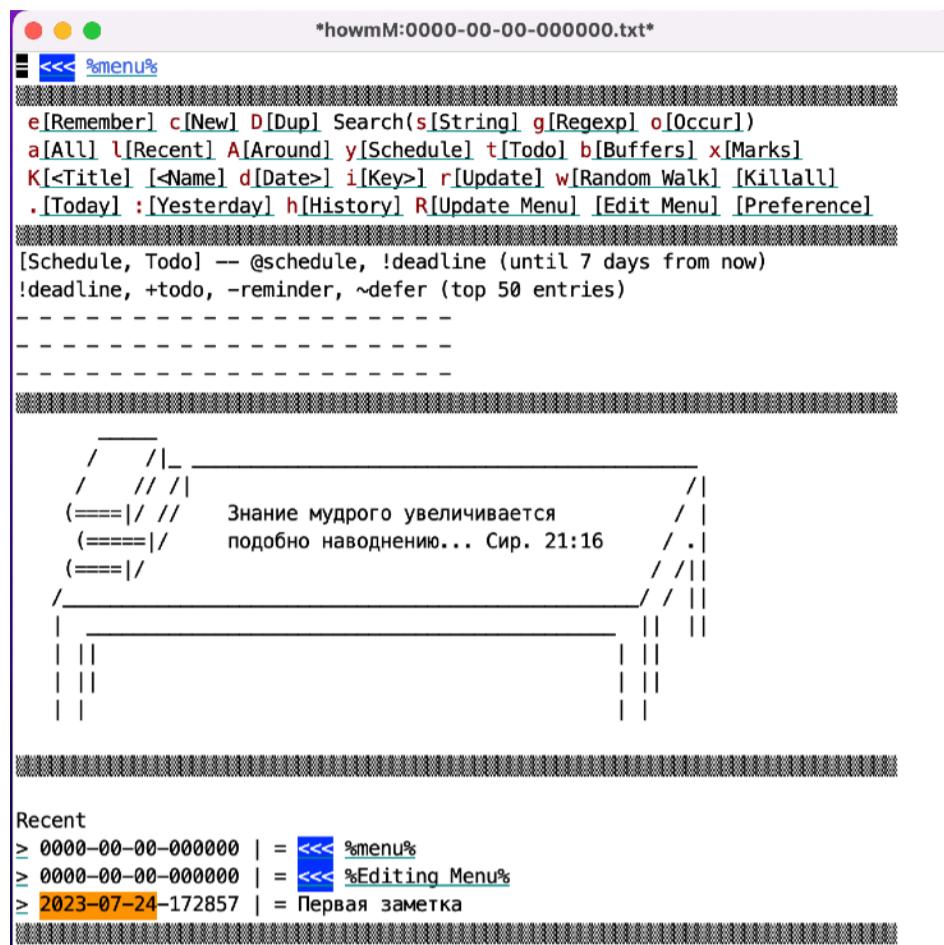


```
[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)
!deadline, +todo, -reminder, ~defer (top 50 entries)
-----
-----
> Sat111 | [2024-03-23]~ упорядочить данные
-----
Выполненных планов - 0 шт.
```

Howm ведет подсчет выполненного за сессию. При новом запуске Emacs результаты обнуляются.

⁷⁹ Не забудьте сделать резервную копию этого файла перед смелыми экспериментами.

Вы можете в дизайнерском угаре сотворить с начальной страницей что-то ужасно прекрасное. Но лучше все же держать себя в руках.



The screenshot shows an Emacs window with the title "*howmM:0000-00-00-000000.txt*". The buffer contains Lisp code for customizing the 'howm' menu. It includes a block of comments about scheduling and todo items, followed by a large, stylized tree diagram made of slashes and dots. At the bottom, there's a 'Recent' section with three entries:

```
e[Remember] c[New] D[Dupl] Search(s[String] g[Regexp] o[0curr])
a[All] l[Recent] A[Around] y[Schedule] t[Todo] b[Buffers] x[Marks]
K[<Title] [<Name] d[Date] i[Key] r[Update] w[Random Walk] [Killall]
.[Today] :[Yesterday] h[History] R[Update Menu] [Edit Menu] [Preference]

[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)
!deadline, +todo, -reminder, ~defer (top 50 entries)

-----
-----
-----
-----
```

/ \ / |
/ / / |
=====| / / | Знание мудрого увеличивается
=====| / | подобно наводнению... Сир. 21:16
=====| / | .
/ |
| |
| |
| |

```
Recent
> 0000-00-00-000000 | = <<< %menu%
> 0000-00-00-000000 | = <<< %Editing Menu%
> 2023-07-24-172857 | = Первая заметка
```

Продвинутые пользователи Emacs могут вытворять и другие трюки благодаря знанию Lisp.



Чтобы увидеть приоритетность каждой задачи в основном меню howm, впишите в файл .emacs следующую строку:

```
(setq howm-menu-todo-priority-format " (%8.1f) ")
```

ВОПРОСЫ И ОПАСЕНИЯ

Вопрос: Не станет ли `howm` тормозить при большом количестве заметок?

Ответ: `howm` не отличается космической скоростью. В тесте с 10000 заметок `howm` думает 26 секунд, прежде чем вывести на экран полный список⁸⁰. Меня это не очень пугает. Работа с картотекой дело неспешное. К тому же, до 10000 реально созданных заметок нужно еще дожить. Если скорость поиска и работы с заметками начинает снижаться, лучше использовать `grep` (см. раздел «Дополнительные настройки» стр. 80).

Вопрос: Не теряются ли важные заметки при отсутствии детализированной структуры?

Ответ: Система позволяет оставаться важным заметкам на плаву. Заметки, которые не слишком интересны, наоборот оттесняются на периферию. Определенная мера хаотичности в картотеке на поверку оказывается выигрышной стратегией. И, наоборот, слишком жесткая классификация и структура обременяют пользователя необходимостью постоянно думать, что и куда разместить.

Вопрос: Почему о `howm` так мало информации? Не подозрительно ли это?

Ответ: Причина преимущественно в языковом барьере. Основной круг пользователей `howm` – это японцы. Плюс, автор программы не сторонник громких заявлений и хайпа.

Вопрос: Эффективнее ли `howm`, чем другие системы управления знаниями?

Ответ: Базовый функционал `howm` аскетичен. Нет таких действий, которые пользователь не смог бы при желании воспроизвести в других приложениях. Достоинство `howm` в разумном балансе простоты и возможностей. Еще одним преимуществом является интегрированность в экосистему Emacs. Если вы стараетесь ограничить себя работой с Plain Text файлами, `howm` будет вполне разумным выбором.

⁸⁰ У меня MacBook 2015 года, и я тестировал скорость `howm` без привлечения `ripgrep`. Со всеми дополнительными оптимизациями скорость стала 20 секунд.

ПРИЛ. №1. СПИСОК ОСНОВНЫХ КОМАНД

Этот список команд следует расценивать как напоминалку. Понять их значимость можно лишь в контексте работы над заметками, где аккорды команд звучат вместе. Здесь представлены лишь команды, специфичные для `howm`, другие полезные клавиатурные сочетания Emacs остались за скобками. В последней колонке указан номер страницы, где конкретная команда упоминается.

Аккорд	Команда	Действие	С
C-c , ,	M-x <code>howm-menu</code>	Вызов основного меню	12
C-c , c	M-x <code>howm-create</code>	Создание новой заметки	16
C-c , e	M-x <code>howm-remember</code>	Создание быстрой заметки	58
C-c , I	M-x <code>howm-create-interactively</code>	Заметка с произвольным именем и местоположением	61
C-c , D	M-x <code>howm-dup</code>	Дублирование заметки	60
C-c , a	M-x <code>howm-list-all</code>	Вызов списка всех заметок	20
C-c , l	M-x <code>howm-list-recent</code>	Вызов списка недавних заметок	23
C-c , .	M-x <code>howm-find-today</code>	Список сегодняшних заметок	24
C-c , :	M-x <code>howm-find-yesterday</code>	Список вчерашних заметок	25
C-c , A	M-x <code>howm-list-around</code>	Записи до и после текущей заметки из списка	23
C-c , K	M-x <code>howm-keyword-to-kill-ring</code>	Копирование заголовка заметки	30
C-u C-c , K	-----	Копирование нумерического имени	31
C-c , s	M-x <code>howm-list-grep-fixed</code>	Поиск и поиск по ключевым словам	41
C-c , g	M-x <code>howm-list-grep</code>	Поиск в том числе с regexp	41
C-c , o	M-x <code>howm-occur</code>	Поиск в теле заметки	55
C-c , h	M-x <code>howm-history</code>	История поиска	43
C-c , i	M-x <code>howm-insert-keyword</code>	Вызов ключевых слов из заметки	38
C-c , p	M-x <code>action-lock-goto-previous-link</code>	В заметке переход к предыдущей ссылке	34
C-c , n	M-x <code>action-lock-goto-next-link</code>	В заметке переход к следующей ссылке	34
C-c , x	M-x <code>howm-list-mark-ring</code>	Список меток-закладок	55
C-c , b	M-x <code>howm-list-buffers</code>	Вызов списка документов из буфера	56
C-c , Q	M-x <code>howm-kill-all</code>	Удалить все буферы (кроме несохраненных)	57

C-c , SPC	M-x howm-toggle-buffer	Переключение между документами howm/не-howm	56
C-c , t	M-x howm-list-todo	Вызов списка дел	72
C-c , T	M-x howm-insert-dtime	Вставка текущей даты и времени	70
C-c , y	M-x howm-list-schedule	Вызов расписания	72
C-c , d	M-x howm-insert-date	Вставка текущей даты	70
C-c , C	M-x howm-create-here	Создание датированного заголовка в длинной заметке	54
C-c , N	M-x howm-next-memo	Переход к следующему заголовку в длинной заметке	53
C-c , P	M-x howm-previous-memo	Переход к предыдущему заголовку в длинной заметке	53
C-c , L	M-x howm-last-memo	Переход к последнему заголовку в длинной заметке	54
C-c , H	M-x howm-first-memo	Переход к первому заголовку в длинной заметке	54
C-c , r	M-x howm-refresh	Активация ссылок-сюда	36
C-c , R	M-x howm-menu-refresh	Обновление основного меню	-
C-c , M	M-x howm-open-named-file	Открытие файла (вне howm)	61
C-c , w	M-x howm-random-walk	Случайное пролистывание заметок	15

ПРИЛ. №2. ОЧ. КР. РУКОВОДСТВО

Данное руководство предназначено для быстрого старта. Вы научитесь создавать заметки, делать перекрёстные ссылки, искать информацию, составлять списки дел.

Этих познаний будет достаточно для заметокоделия, ведь в конечном итоге все сводится к созданию карточек и линкованию идей. Однако, если вы хотите понастоящему овладеть howt, чтения расширенного руководства не избежать.

Ниже пошаговая инструкция овладения базовыми функциями howt.

Шаг первый: создаем несколько заметок

Откройте Emacs

Используйте клавиатурный аккорд: C-c , c

Перед вами откроется шаблон заметки howt, который будет выглядеть вот так:

```
=  
[2023-05-13 11:12]
```

После знака равно вы можете написать заголовок заметки. После даты лучше оставить одну пустую строку и затем создать заметку.

Сохраните запись командой C-x C-s

Создайте вторую заметку, нажав еще раз C-c , c

Заметьте, что шаблон будет отличаться. Теперь в нем напротив даты есть ссылка на первую заметку.

Напишите новую заметку и сохраните результат - C-x C-s

Выходите из Emacs - C-x C-c

Шаг второй: учет заметок

Откройте Emacs

Используйте клавиатурный аккорд C-c , ,

Так вы попадете в основное меню howt. Здесь вам доступны основные команды, список дел (о нем чуть дальше) обзор недавних заметок, список случайных заметок.

Нажмите клавишу `a`, чтобы увидеть список созданных вами заметок. Nowt разделяет фрейм на два окна. В верхнем находится список, в нижнем – предпросмотр содержания.

Передвигайте курсор с списке заметок клавишами `p` и `r`. Вы увидите, как в нижнем окне меняется содержание.

Чтобы пролистывать заметку в режиме предпросмотра, нажмите `space` или `backspace` (`delete` на MacBook).

Чтобы перейти в заметку, разместите курсор на элементе списка и нажмите `RET`.

Вы можете вызывать список не только из основного меню.

Клавиатурный аккорд `C-c , a` вызовет полный список. Если вам нужен список недавних заметок? нажмите `C-c , l`

Шаг третий: ищем заметки

Откройте Emacs.

Используйте клавиатурный аккорд `C-c , s`

Ведите слово, которое вы хотите найти в заметках и нажмите `RET`. Вы увидите знакомый интерфейс: в верхней части список, в нижней – предпросмотр.

Шаг четвертый: линкуем заметки

При открытом Emacs используйте клавиатурный аккорд `C-c , a`

Выберите из списка заметку, которую вы хотите заликовать, и перейдите в нее. Скопируйте заголовок заметки, используя аккорд: `C-c , K`

Можно скопировать нумерическое имя заметки: `C-u C-c , K`

Вернитесь к списку: `C-c , a`

Выберите заметку, в которой будет размещаться ссылка, и перейдите в нее.

Наберите `>>>` и вставьте ссылку командой `C-y`

Сохраните результат: `C-x C-s`

Наведите курсор на ссылку и нажмите `RET`.

Nowt откроет интерфейс из двух окон. В верхнем будет список (заметка на которую вы сослались и заметка со ссылкой). В нижнем – предпросмотр. Адресную заметку можно узнать по знаку равно рядом с заголовком. Если вы используете

нумерическое имя (что предпочтительнее), адресная заметка будет первой в списке.

Шаг пятый: создаем ключевые слова

Создайте новую заметку или откройте существующую.

Наберите <<<

Разместите после этого термин, который станет активной ссылкой во всех заметках. Путь это будет «карточка».

<<< картотека

Сохраните результат.

Перейдите в другую заметку и наберите слово картотека. Сохраните заметку. Заметьте, что слово «карточка» окажется подчеркнутым.

Наведите на него курсор и нажмите RET.

Nowt откроет интерфейс из двух окон. В верхнем будет список всех заметок, где встречается слово «карточка». В нижнем – предпросмотр.

Самой первой в списке будет заметка, в которой вы оставили «ссылку-сюда».

Шаг шестой: управляем списком дел

Откройте заметку или создайте новую.

Наберите планируемую дату в квадратных скобках – [2023-08-30]

Поставьте после закрывающей скобки значок, который определит тип записи.

[2023-08-30]- Памятка (практиковать осознанность)

[2023-08-30]+ Задача (починить табуретку)

[2023-08-30]@ Событие (сходить на концерт)

[2023-08-30]! Дедлайн (отправить статью в следующий выпуск журнала)

[2023-08-30]~ Откладывалка (вспоминать каждый месяц о радости бытия)

Сохраните документ – **C-x C-s** – и перейдите в основное меню. Вы увидите запланированное мероприятие под панелью с командами.

Nowt формирует список задач, событий и т.п. так, чтобы наиболее насущные оказались в первых строках.

Разместите курсор на значке > перед записью и нажмите RET. Nowt откроет соответствующую заметку.

Чтобы завершить задачу/событие, разместите курсор на значке после квадратных скобок и нажмите RET два раза.

Запись изменится на следующую:

[2023-08-30]. [2023-08-30]- Памятка

Первая дата время окончания, вторая - изначальный план.

Завершенные дела исключаются из списка на главной странице.