



# 一人お手軽 Wiki もどき Заметкоделие в Emacs использование пакета HOWM

# ОГЛАВЛЕНИЕ

Вступление .....	2
История создания howm .....	3
Клавиатурные сочетания в Emacs .....	5
Установка пакета howm .....	8
Основное меню howm .....	12
Создание заметок .....	16
Работа со списками .....	19
Сортировка заметок в списках .....	25
Работа со ссылками .....	28
Ключевые слова в howm .....	34
Функции поиска и фильтрации .....	40
Удаление заметок .....	46
Сшивание заметок .....	48
Работа с длинными заметками .....	51
Работа с буферами .....	55
Быстрые заметки .....	57
Дублирование .....	59
Неформатные заметки .....	60
Управление делами в howm .....	61
Дополнительные настройки .....	69
Вопросы и опасения .....	77
Прил.№1. Список основных команд .....	78
Прил.№2. Оч.кр.руководство .....	80

# 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, 2023

# 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 – «Вики, дружественная к человеку» или «Удобная персональная вики».

Пакет `howm` удивляет компактностью и элегантностью решений. В данном случае характерно, что это сделано в Японии. Автор – Казуюки Хираока – является преподавателем в университете Миядзаки на Инженерном факультете (Кафедра проектирования информационных систем). 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. “Запарность” – *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 файлами<sup>1</sup>. 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.)<sup>2</sup>.

---

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

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

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

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

`Howt` следует схожему принципу. Категории и разделы отсутствуют<sup>3</sup>. 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 `howt` была вдохновлена разными приложениями. Среди них Q-Pocket, ChangeLogMemo, HyperCard, HashedWiki и др.

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

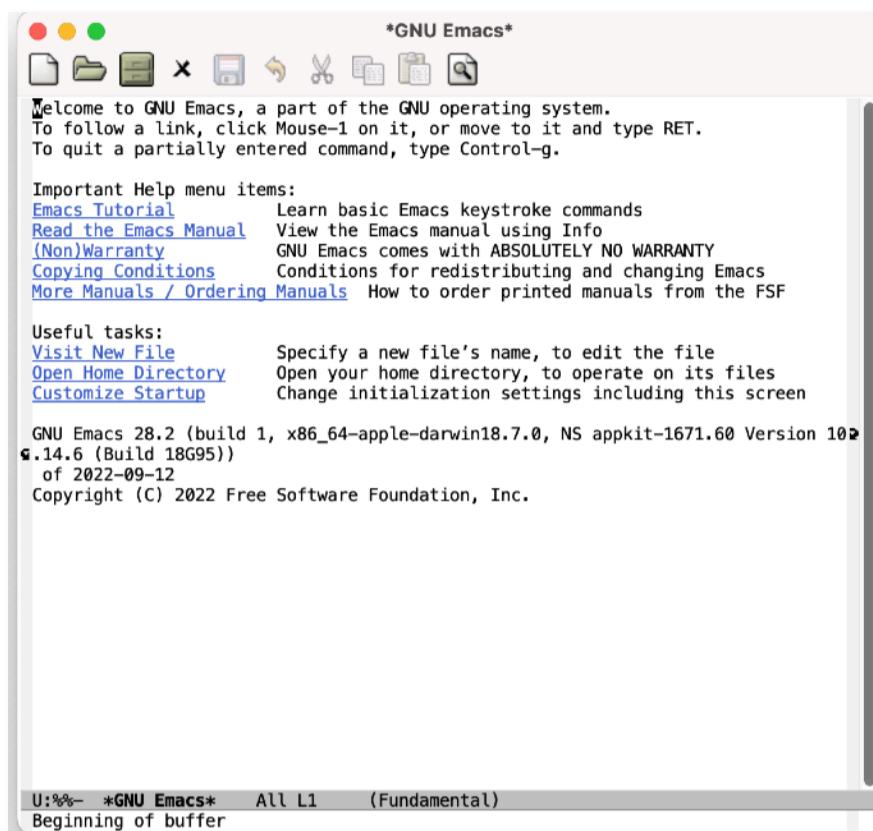
Мы перейдем к обсуждению возможностей `howt` на стр. 12, но перед этим мне необходимо сделать пару вводных замечаний об использовании Emacs, установке новых пакетов и т.п.

---

<sup>3</sup> 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.

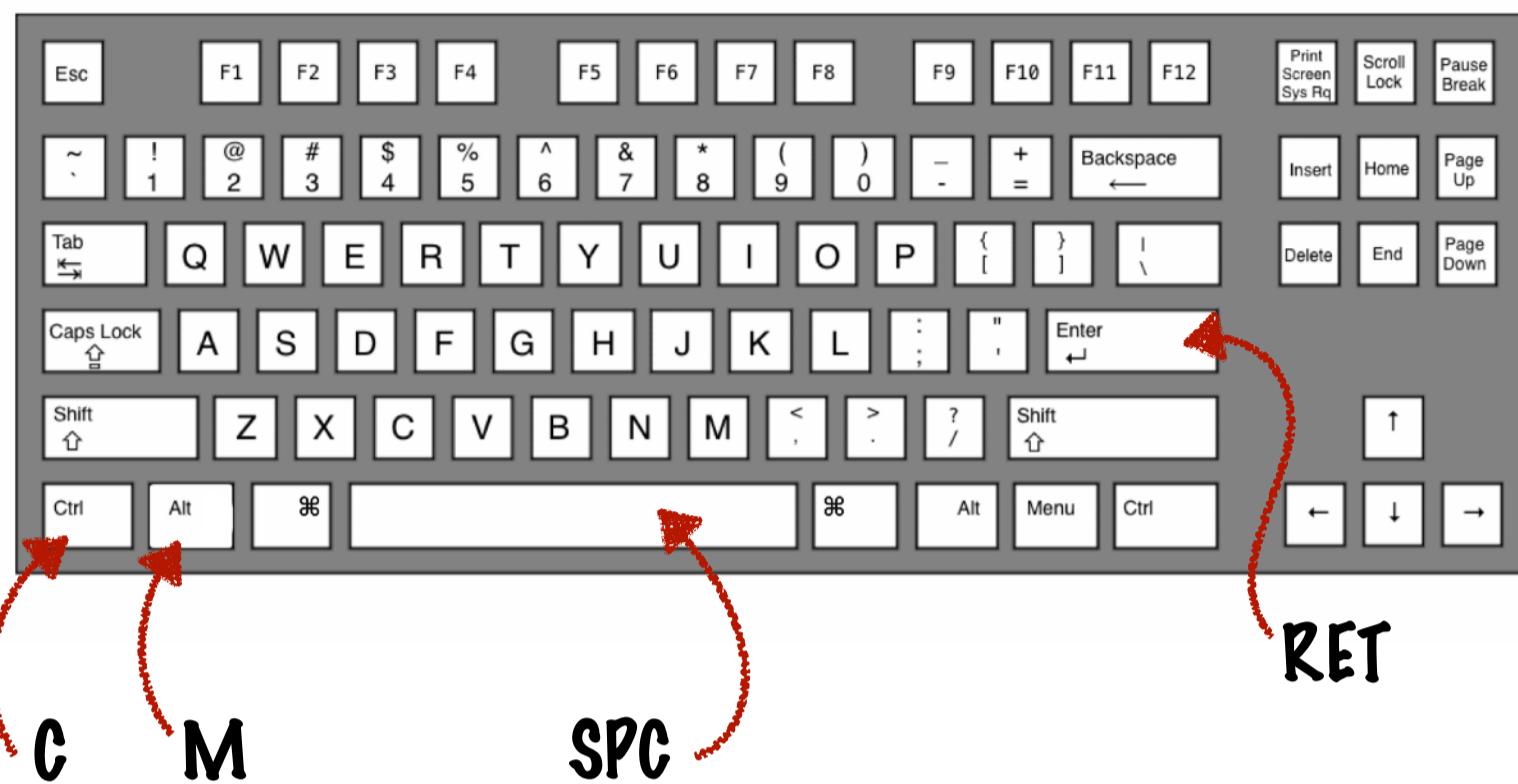
Одной из важных особенностей Emacs является «клавиатуроцентричность». Абсолютно любое действие можно совершить, не покидая своими пальцами клавиатуры. «Долой мышиную возню с текстом» – таким мог бы быть один из лозунгов Emacs.

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

В следующих разделах пособия мы рассмотрим команды, используемые в howm<sup>4</sup>.

<sup>4</sup> 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**<sup>5</sup>.

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 используется клавиша **Alt**, а традиция обозначать ее буквой **M** осталась.

<sup>5</sup> 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 прекрасным текстовым редактором. Множество замысловатых манипуляций с текстом доступны вам прямо после его установки.

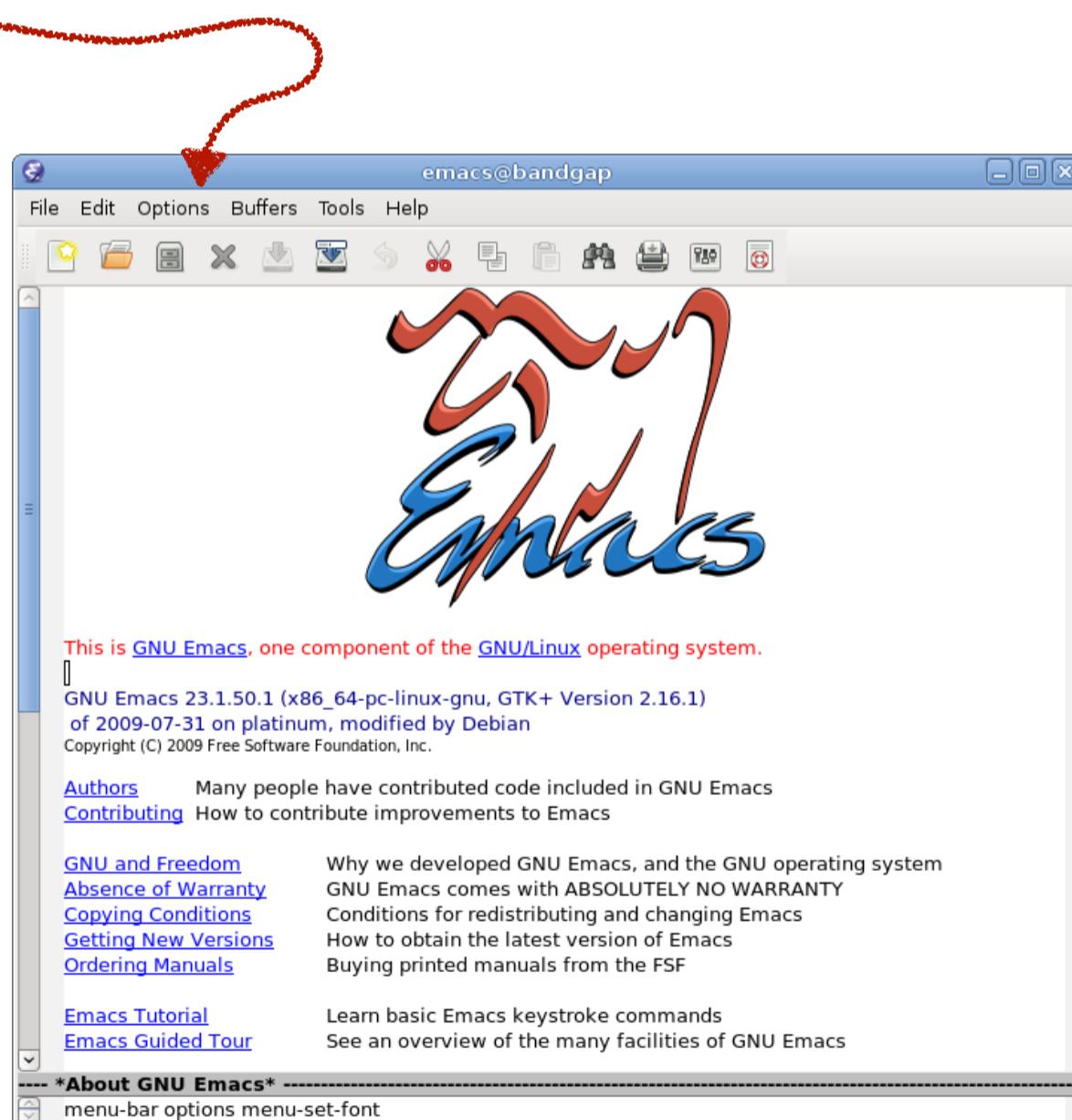
Однако возможности Emacs могут быть адаптированы под ваши нужды благодаря дополнительным пакетам.

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 опциями<sup>6</sup>.



<sup>6</sup> 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 Archive (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 Archive. 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 ним 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. Наберите `C-x d`

2. Дальше введите `~/.emacs` и нажмите RET

В открывшемся файле<sup>7</sup> ниже имеющихся записей разместите строку:

`(require 'howm)`

Обратите внимание, что апостроф перед словом `howm` должен быть прямым. Простой орфографический ‘ не подойдет.

Этих действий будет достаточно для исправной работы пакета<sup>8</sup>.

---

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

<sup>8</sup> 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 судна<sup>9</sup>.

Вся информация в окно не влезает. Пролистывать вниз можно, нажимая на пробел (или `C-v`), вверх – кнопка `delete` (или `M-v`).

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

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`.

---

<sup>9</sup> 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`.

Развернутый вид на начальную страницу howt выглядел бы так:

Раздел меню с основными командами. Доступ к этим командам возможен также через горячие клавиши.

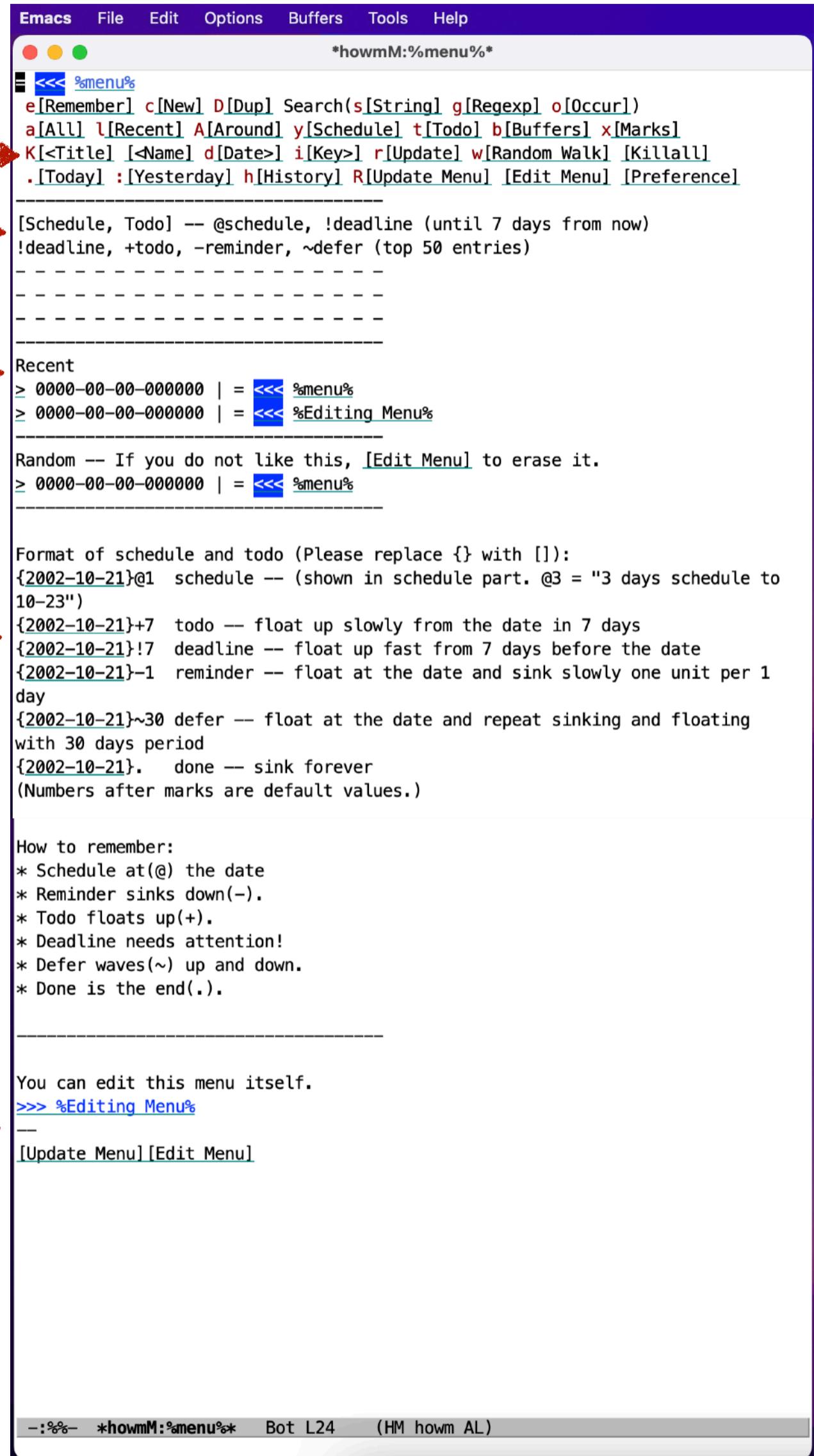
Раздел со списком дел, задач, дедлайнов. Nowm включает довольно гибкую систему организации и планирования.

Здесь отображаются заметки которые вы недавно создали или редактировали.

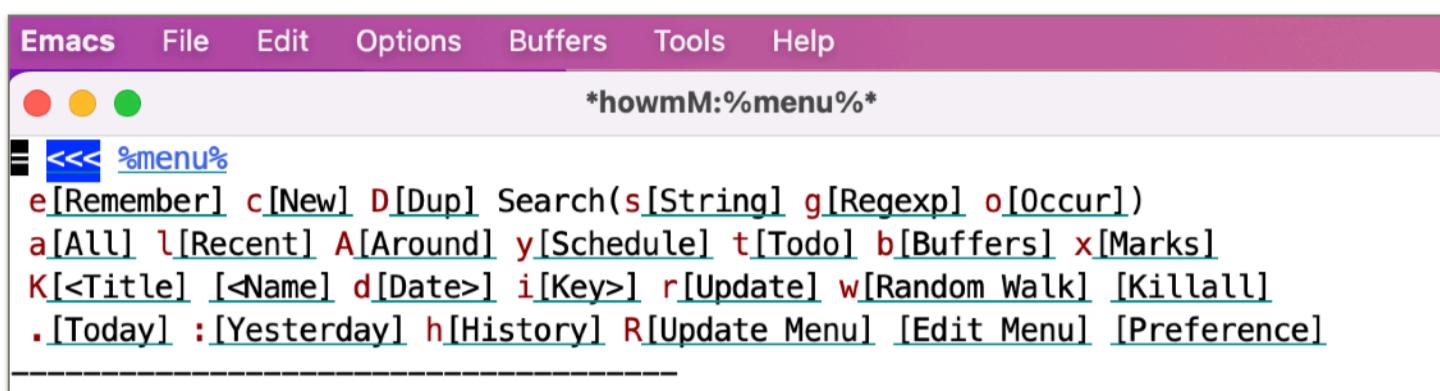
Здесь будет список  
случайных заметок.  
Возможность наткнуться  
на что-то подзабытое.

## Памятки о синтаксисе для создания списка дел.

## Возможность редактировани я меню.



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** – поиск по ключевым словам.

**r** – обновление заметок, в которых есть «ссылки-сюда». Если в каком-то тексте появились ключевые слова, они будут активированы<sup>10</sup>.

**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`<sup>11</sup>. Просто нужно будет предварить символ, указанный в меню, следующим клавиатурным сочетанием: **C-c ,**

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

---

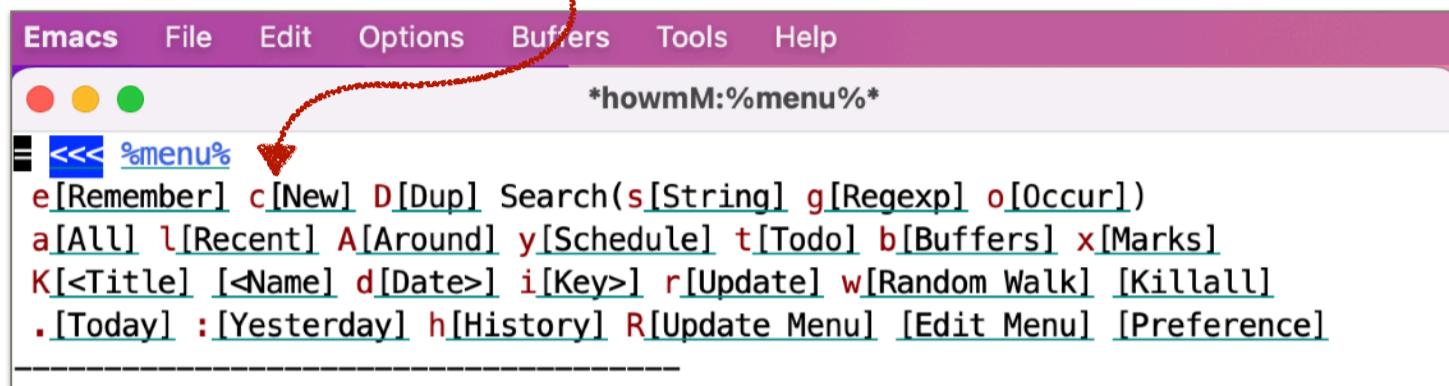
<sup>10</sup> Подробнее об этом в главе «Ключевые слова в `howm`».

<sup>11</sup> `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.

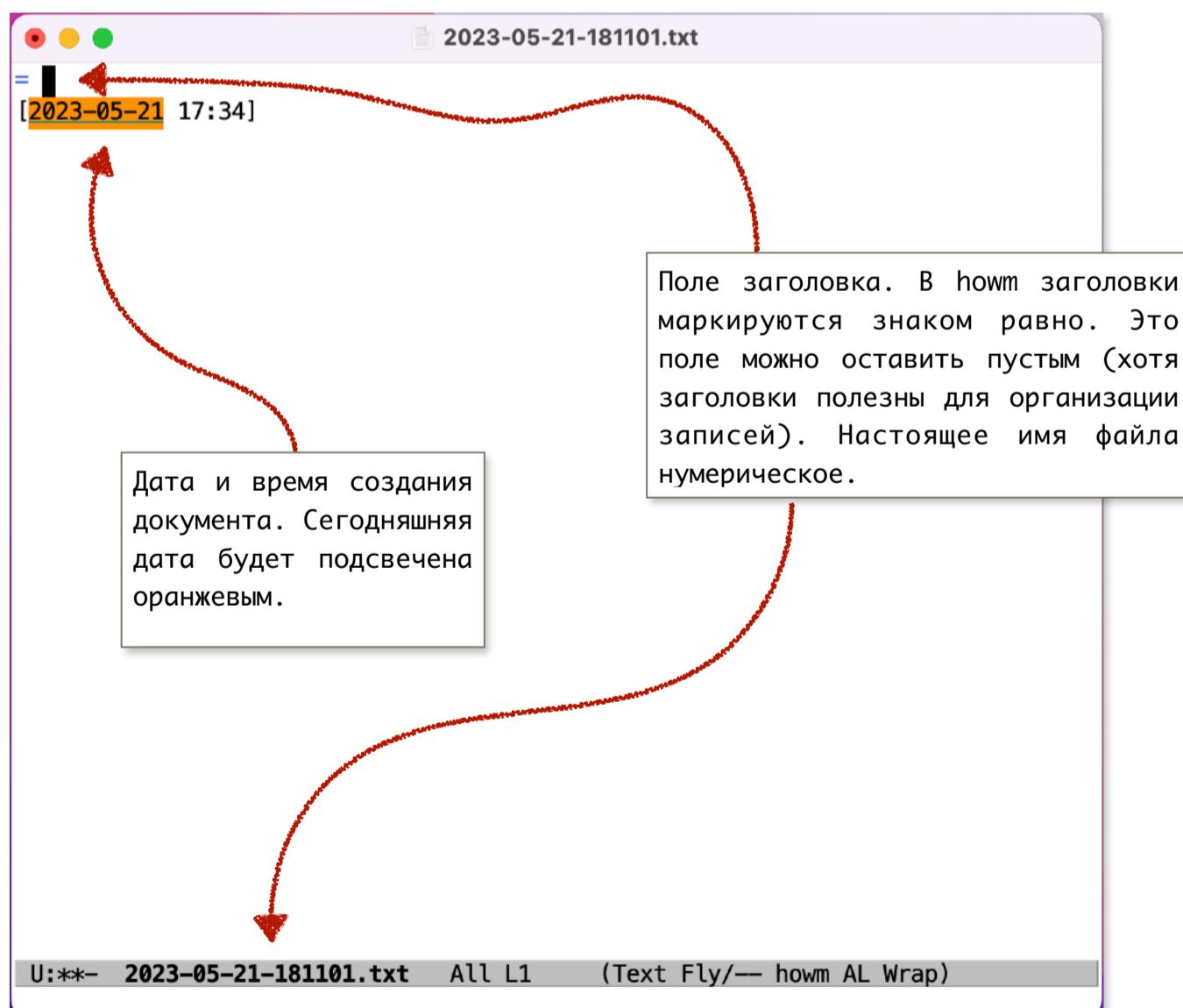
Если вы в основном меню howm, достаточно переместить курсор на соответствующую кнопку [New] или просто нажать клавишу **c**. Она, понятное дело, соответствует начальной букве слова *create*.



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?



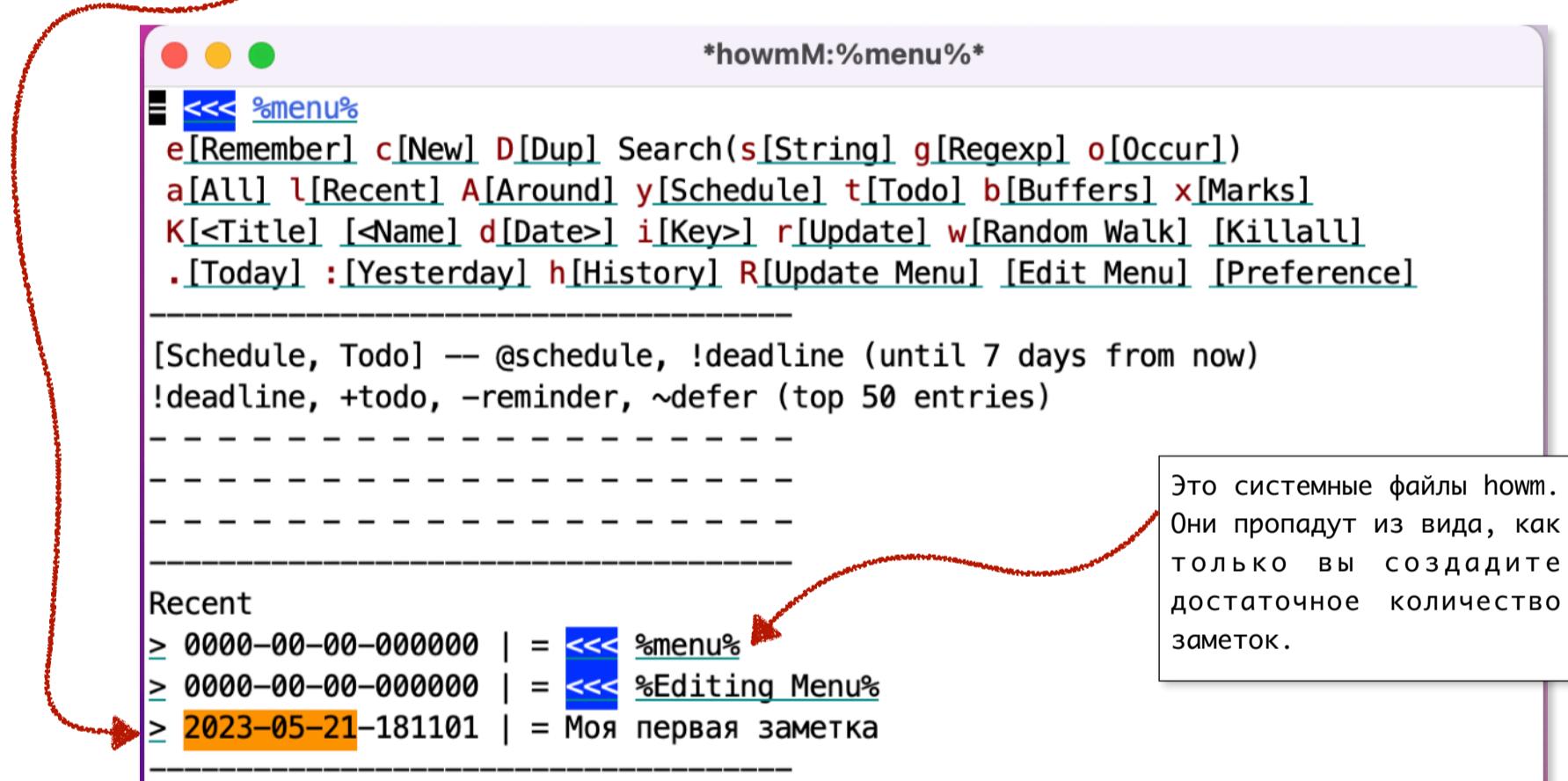
Уникальным идентификатором заметки является нумерическое имя. Поэтому, заголовок может быть идентичным у разных заметок (хотя, это плохая практика) или вовсе отсутствовать.

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.) невозможны<sup>12</sup>.

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 навести курсор на символ `>` рядом с заметкой и нажать RET.

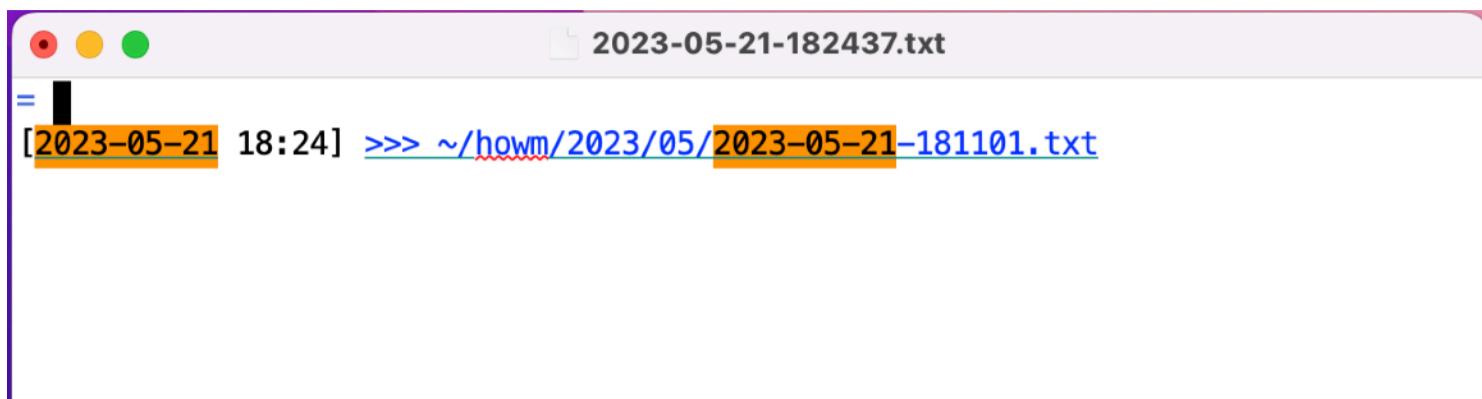
В процессе написания первой заметки вас могут посетить идеи, развивающие тему.

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`

<sup>12</sup> Вы можете поменять расширение файла (см. раздел «Дополнительные настройки», стр. 71). 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 запись<sup>13</sup>.

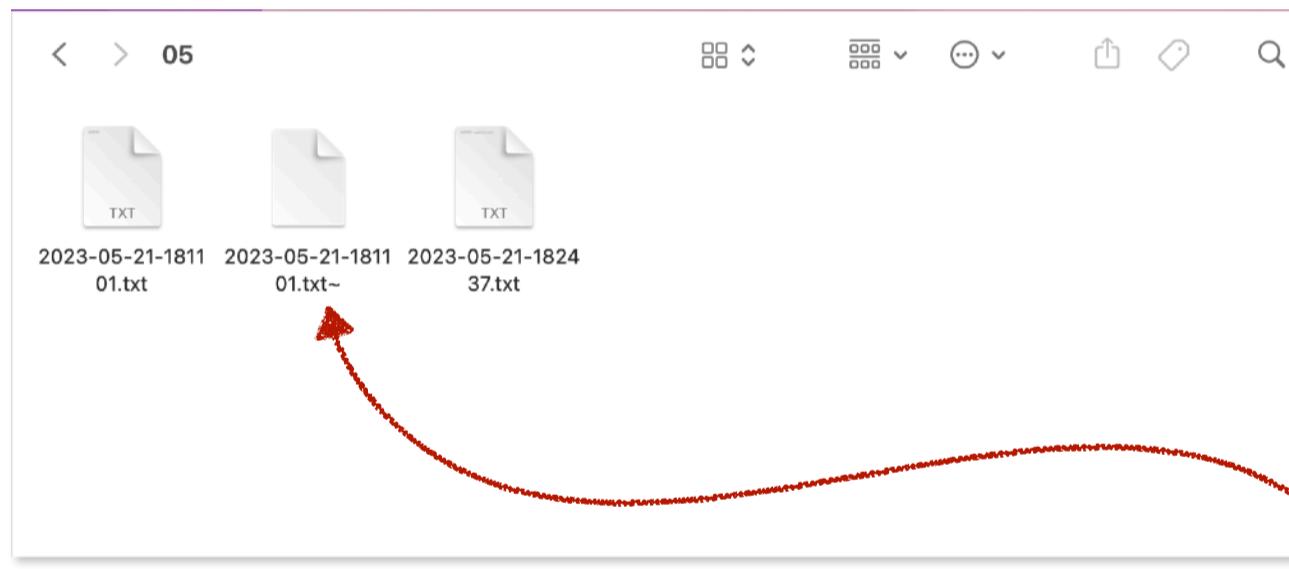


Всякий раз, когда вы, находясь в одной заметке, создаете другую, howm их автоматически линкует<sup>14</sup>. Подручное средство для ассоциативного мышления.

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 месяца<sup>15</sup>.



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.

<sup>13</sup> О том, как работают ссылки в howm разговор пойдет чуть дальше.

<sup>14</sup> Ссылка появляется только во второй заметке.

<sup>15</sup> 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)

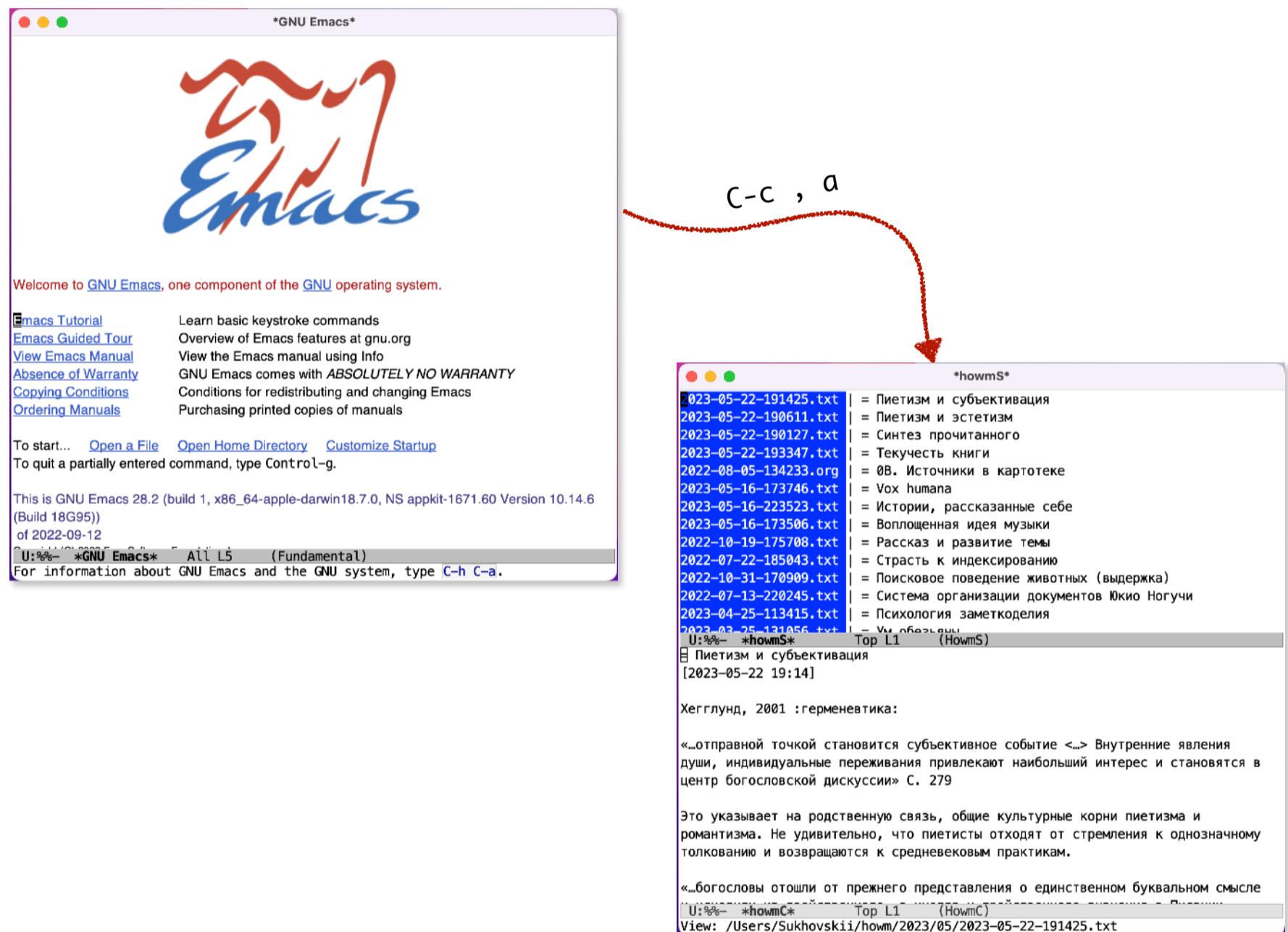
Поскольку списки – это ключевой элемент в работе с `howm`, стоит подробно рассмотреть функции и возможности, скрытые здесь. Какой бы список пред вами ни был, принципы работы с ним будут одинаковыми.

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`



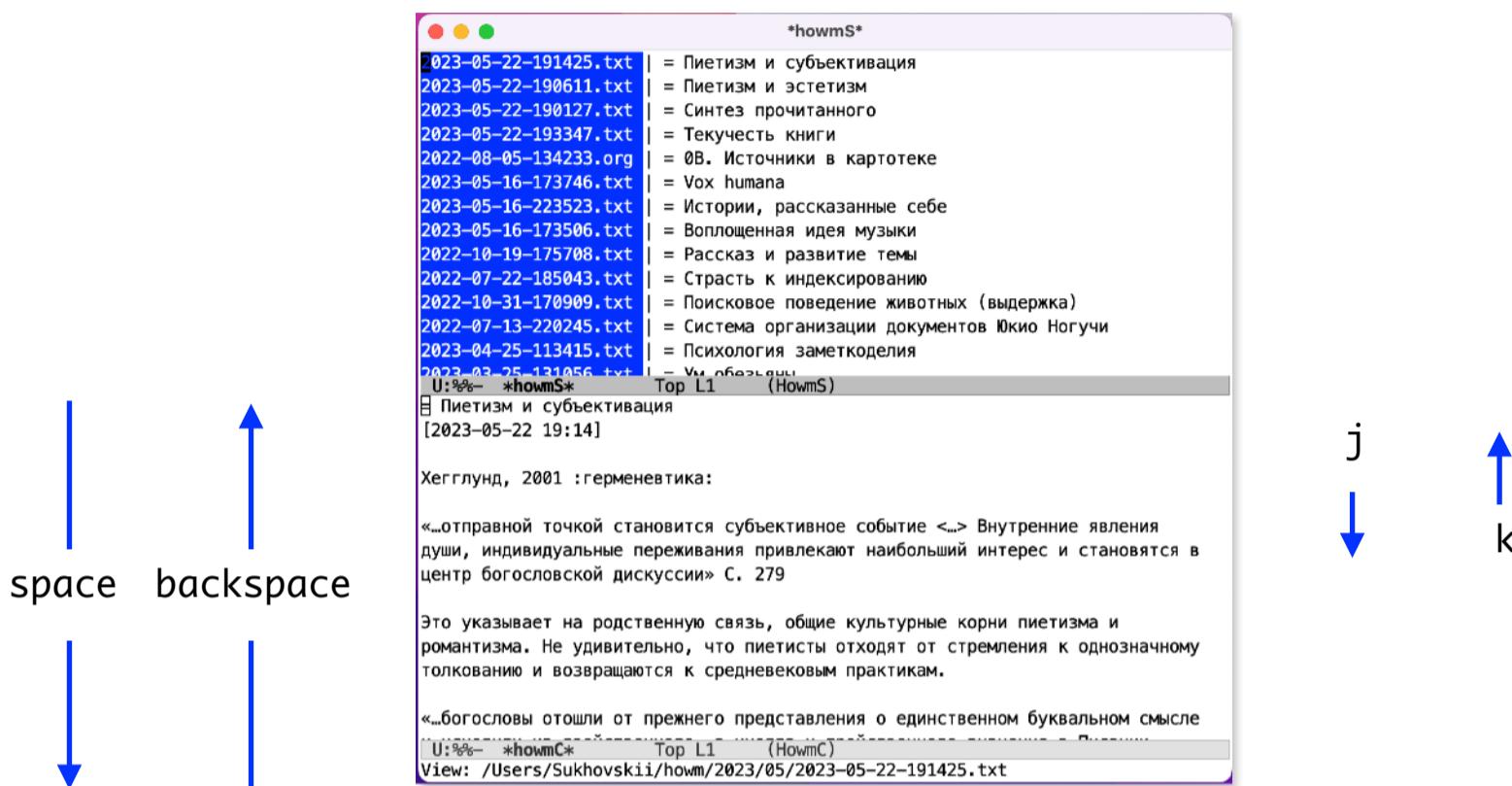
Поговорим сначала о командах навигации.

Перемещение курсора по списку осуществляется клавишами `p` и `r`, вниз и вверх соответственно<sup>16</sup>. В нижнем отсеке по мере вашего движения перелистываются

<sup>16</sup> 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 backspace<sup>17</sup>. 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 RET<sup>18</sup>.

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 список<sup>19</sup>. Чтобы вернуть предпросмотр, нажмите 2. Вы также можете переключаться туда-сюда, нажимая v.

<sup>17</sup> На Macbook это клавиша delete.

<sup>18</sup> Можно также перейти в заметку и нажать С-x 1. Это стандартная для Emacs команда, которая закрывает все окна, кроме текущего.

<sup>19</sup> 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. 72

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 работали<sup>20</sup>.

Режим общего списка позволяет вам осуществить несколько полезных операций.

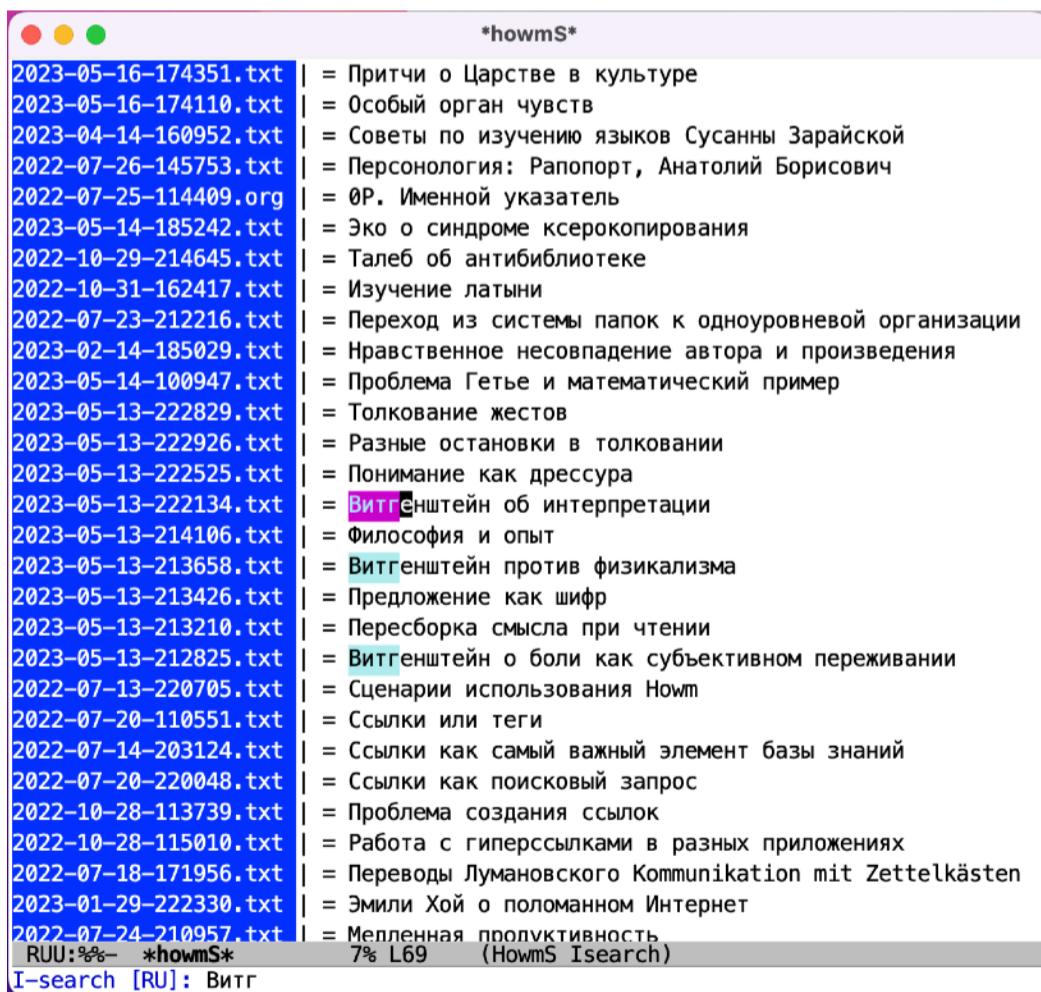
## Подсчет заметок

Наберите команду подсчета строк - `C-x l`, чтобы узнать, сколько заметок в вашей картотеке. Цифра появится в нижнем минибуфере<sup>21</sup>.

## 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`.



<sup>20</sup> Шанс, что вам потребуются именно они очень высок. Именно на этом построена система самоорганизации Юкио Ногучи.

<sup>21</sup> 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 дней<sup>22</sup>.

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" выпиской и выдержкой». С. 691 Сочинения

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

<sup>22</sup> Заметьте, именно за 7 дней. Если вы сделаете запрос в четверг, то в списке будут все измененные/созданные заметки с прошлого по этот четверг.

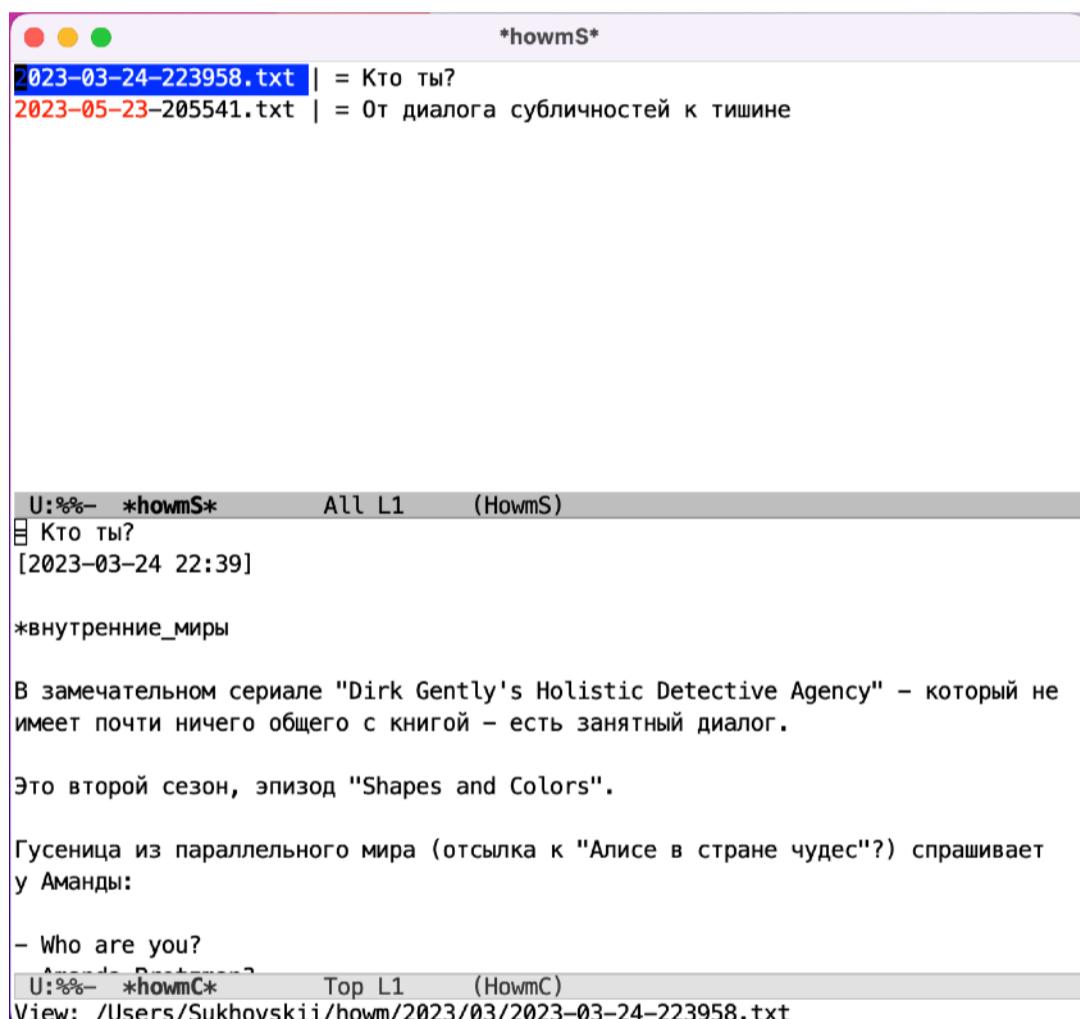
Что если на склоне дня вы захотели посмотреть все заметки, которые сегодня были внесены в базу? Конечно, они и так подсвечены оранжевым.

Однако, если вам хочется отдельный список, используйте команду С-с , .

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

Если вы используете нумерические ссылки<sup>23</sup>, 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 фоне.



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 видите, взаимодействие со списками – это центральный способ работы в howm.

<sup>23</sup> О них речь пойдет дальше.

# СОРТИРОВКА ЗАМЕТОК В СПИСКАХ

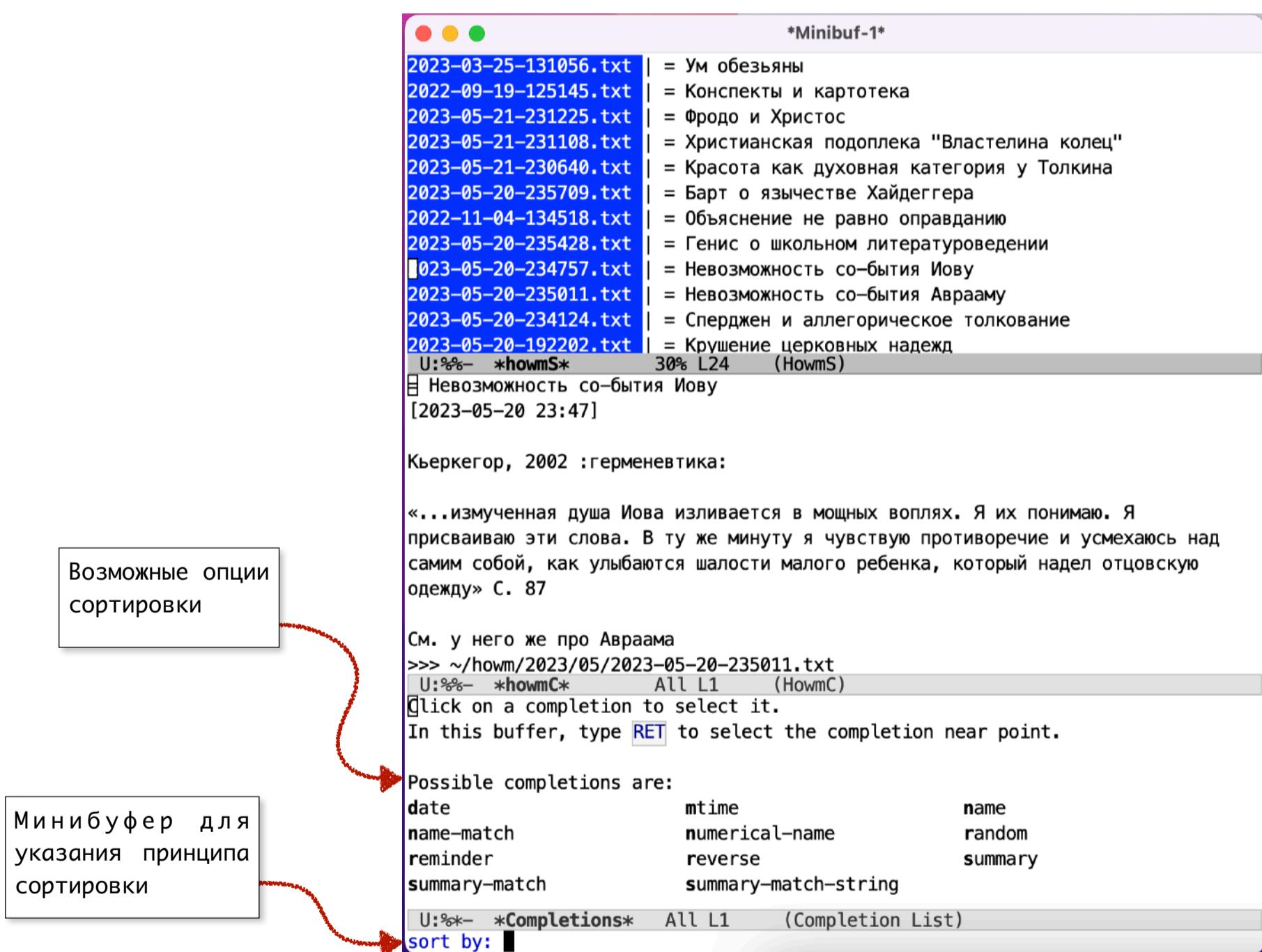
Запросив у howm список заметок командой `C-c , a` или `C-c , l`, вы получаете его рассортированным в порядке свежести правок. То есть наверху будут заметки, с которыми вы недавно работали.

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 варианты<sup>24</sup>.



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

<sup>24</sup> 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 заметка<sup>25</sup>.

## **name-match**

Благодаря этой команде, вы можете расположить заметки в удобной для вас очередности<sup>26</sup>. Howm попросит вас указать имя файла, который необходимо

---

<sup>25</sup> Напомню, что на начальной странице howm уже есть список случайных заметок.

<sup>26</sup> 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 фильтров<sup>27</sup>.

### **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 буквы<sup>28</sup>.

### **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 клиентом<sup>29</sup>.

---

<sup>27</sup> Использование Regex - это отдельная тема, выходящая за пределы данного краткого пособия.

<sup>28</sup> Вам даже не обязательно забрасывать инородные файлы в howm. Вы можете просто указать путь к ним. См. раздел «Дополнительные настройки», стр. 72

<sup>29</sup> Если вы используете почтовый клиент, такой как Wanderlust, можно настроить howm на поиск файлов с соответствующей директории. См. раздел «Дополнительные настройки», стр. 72.

# 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.

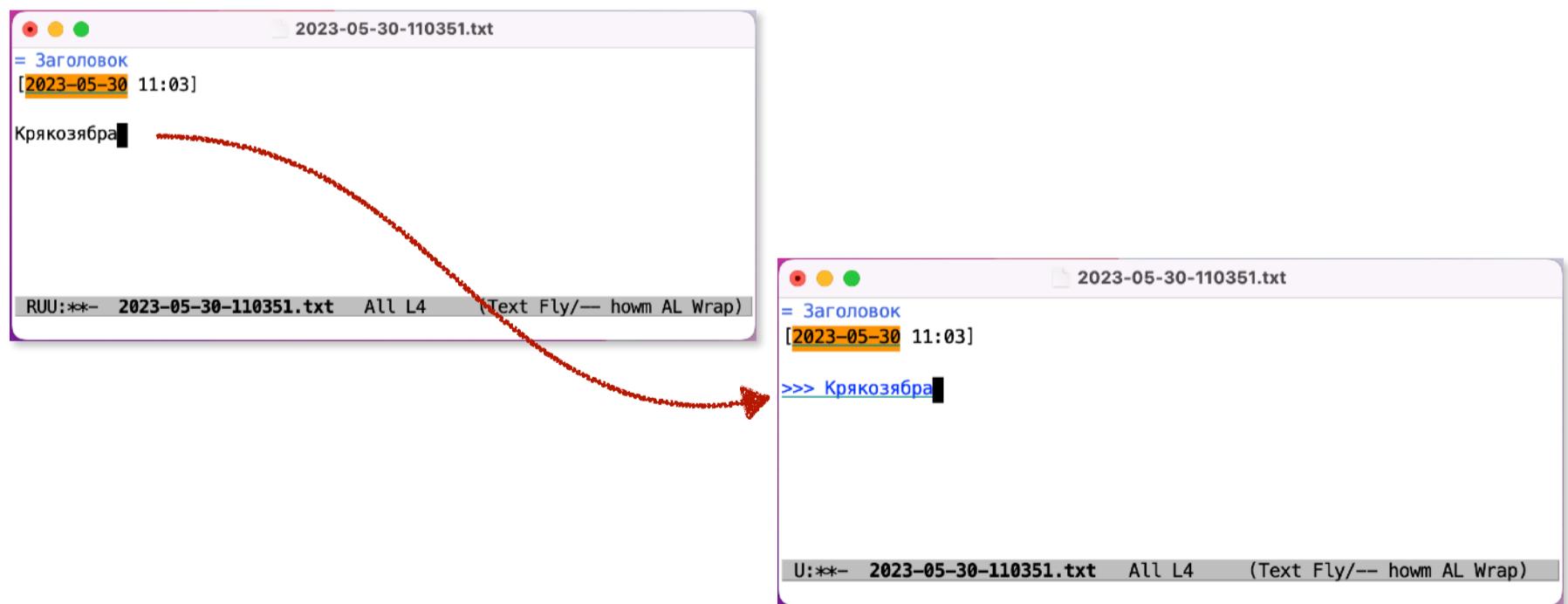
>>> - ёлочка, повернутая вправо.

<<< - ёлочка, повернутая влево<sup>30</sup>.

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 ссылку<sup>31</sup>.



<sup>30</sup> Есть также возможность использовать традиционные для Markdown квадратные скобки. Об этом чуть дальше.

<sup>31</sup> 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:

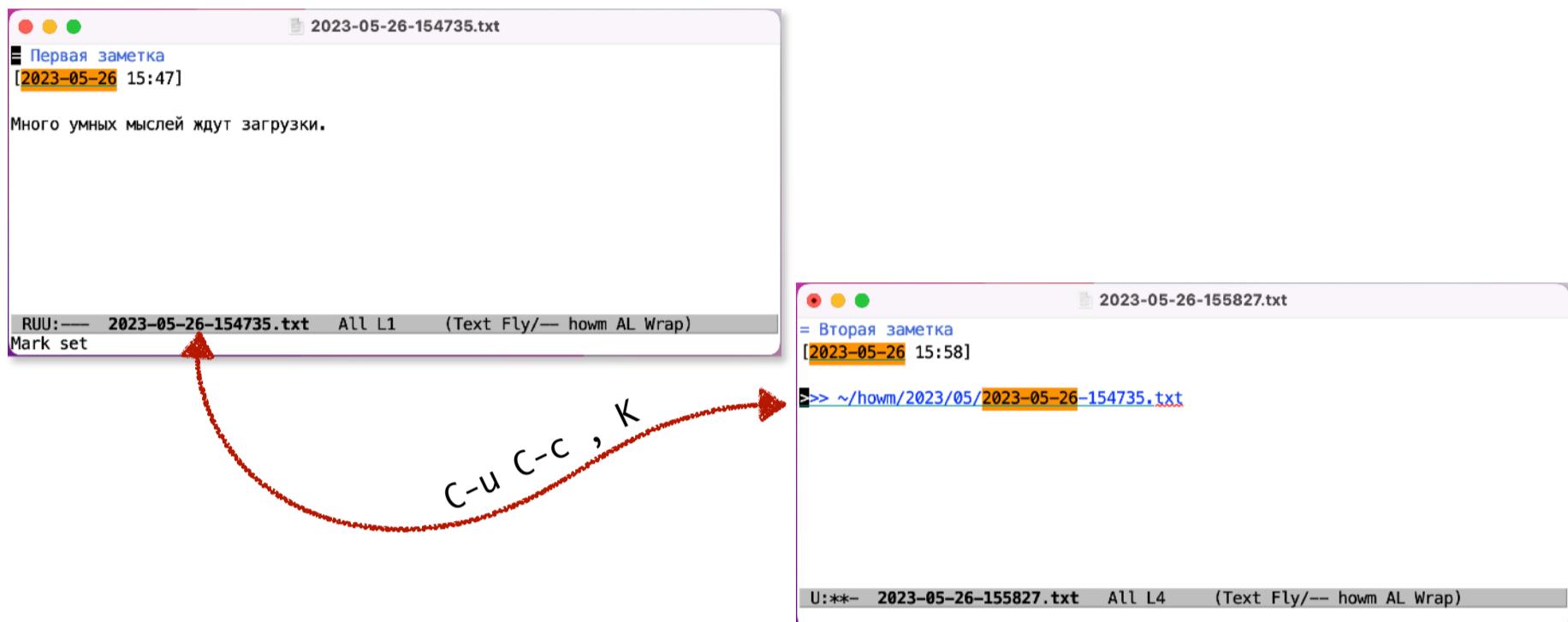
- Top Left Window:** Title bar: 2023-05-26-154735.txt. Content: '= Первая заметка [2023-05-26 15:47]'. Below it: 'Много умных мыслей ждут загрузки.' A red arrow points from the text '= Первая заметка' to the right, indicating where to type a link.
- Top Right Window:** Title bar: 2023-05-26-155827.txt. 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.
- 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:** '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 after the right-pointing colon.)
- Right Annotation:** 'Можно вписать от руки, но легче скопировать и вставить.' (You can type by hand, 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.)
- Bottom Annotation:** 'Вас перебросит в двухоконный режим (список и окно предпросмотра).' (You will be redirected to a two-window mode (list and preview window).)
- Bottom Annotation:** 'Вы увидите заметку со ссылкой >>>' (You will see a note with the link >>>)
- Bottom Annotation:** 'И заметку с определенным заголовком (она со знаком равно).' (And a note with a specific title (it has an equals sign).)
- Bottom Annotation:** '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.)
- Bottom Annotation:** '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. These can be links or simple 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 сверху<sup>32</sup>. 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 другую<sup>33</sup>.

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

<sup>32</sup> В случае со ссылкой на заголовок список формируется иначе и адресная заметка может оказаться посередине или в конце.

<sup>33</sup> 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`

file://~/howm/2023/05/2023-05-26-154735.txt

Лучше, как мне кажется, привыкнуть к ссылочной системе howm (ссылка-список-переход). Она обладает целым рядом преимуществ при работе с заметками<sup>34</sup>.

Продумайте также, когда использовать в качестве ссылок нумерическое имя файла, а когда заголовок. Желательно, однако, сохранять последовательность<sup>35</sup>.

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).

Howm распознает обычные ссылки на интернет страницы. Любая ссылка, скопированная из адресной строки браузера, будет рабочей<sup>36</sup>.

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

Дальше необходимо указать полный путь к файлу. Например:

`file://~/Organon/0A.Ten_minutes.org`

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.

<sup>34</sup> Прежде всего, это возможность напоминать себе о других траекториях в ссылочной системе.

<sup>35</sup> 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.

<sup>36</sup> Emacs позволяет просматривать интернет-страницы. Однако для этого необходимы дополнительные настройки. По умолчанию ссылка откроется в вашем штатном браузере.

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 ссылку<sup>37</sup>.

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 текстами<sup>38</sup>.



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`

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 заликовать, и скопировать адрес одной в другую. Избалованным автоматизацией людям это может показаться ужасно медленным процессом.

---

<sup>37</sup> 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.

<sup>38</sup> В разделе «Дополнительные настройки» (стр. 74) можно узнать, как подружить режимы `howt` и Org.

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?

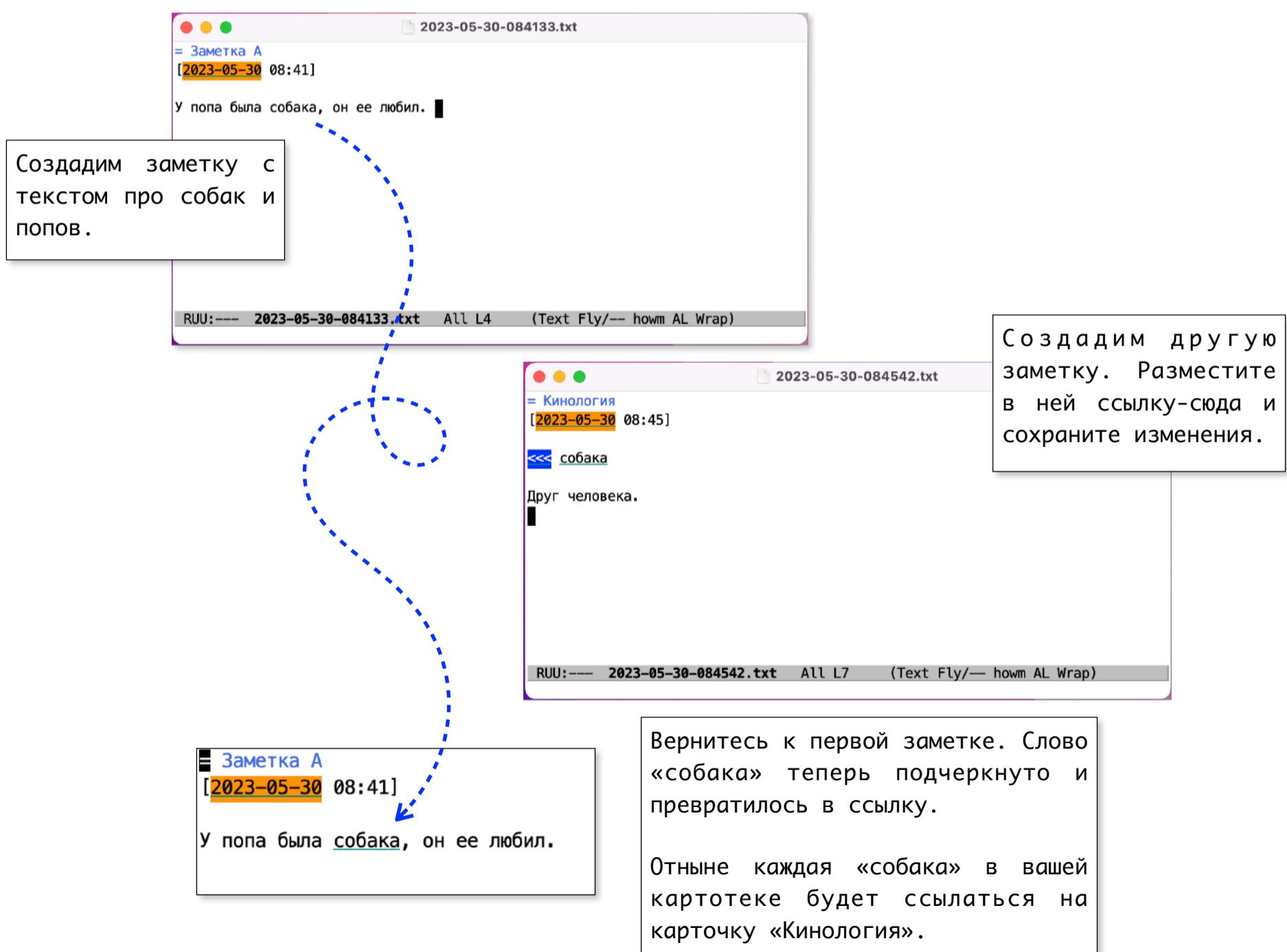
Она тоже создает ссылку, но с другими свойствами. Можно назвать это «ссылка-сюда», а предыдущий тип – «ссылка-отсюда».

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

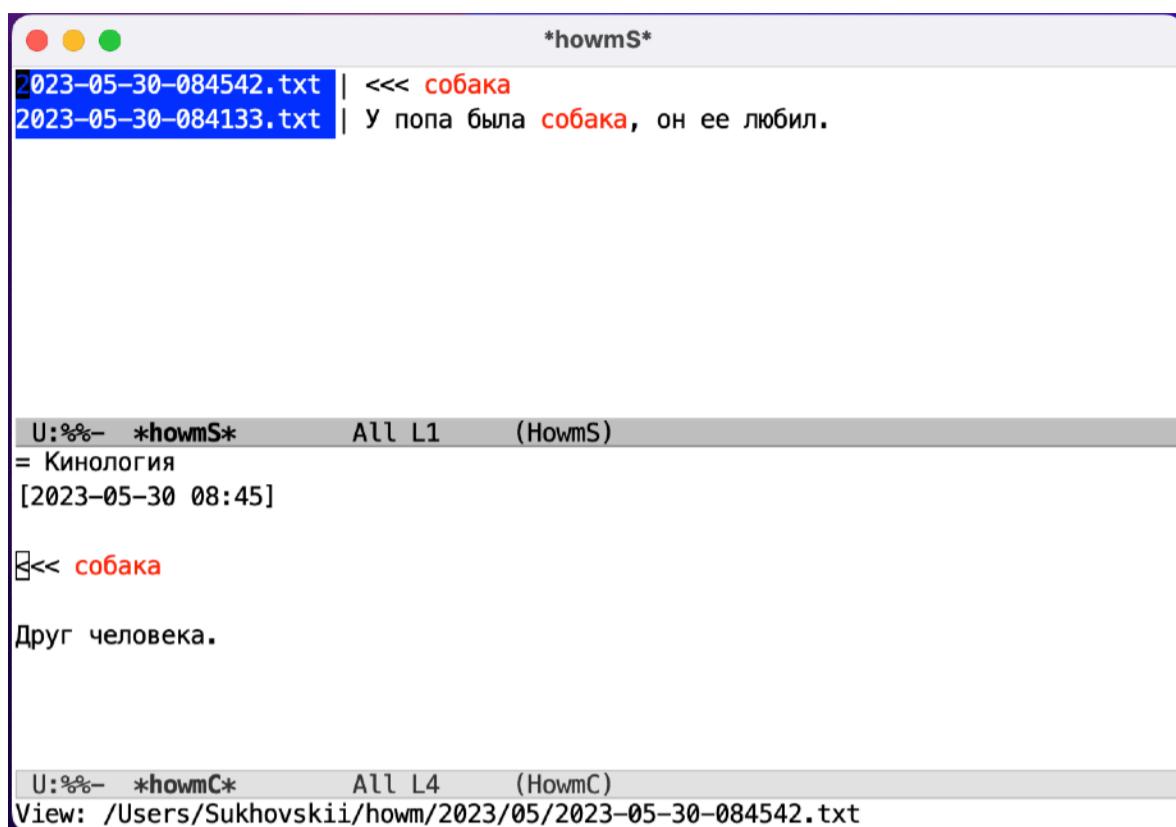
Ёлочка-влево <<< имеет прямо противоположный смысл. Вы не хотите никуда переходить. Вы хотите, чтобы все другие заметки с ключевым словом ссылались сюда.

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 критериям<sup>39</sup>.

Поскольку заметка с ключевым словом всегда будет сверху списка, остается нажать RET, чтобы в нее перейти<sup>40</sup>.

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.
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 С-с , r. Данную команду можно также запустить из начального меню howm (там это клавиша r).

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.

<sup>39</sup> Подробнее о поиске в следующей главе.

<sup>40</sup> Разместив курсор на подчеркнутом слове и нажав RET два раза, вы сразу перемещаетесь в адресную заметку.

В русском языке неизбежно возникает проблема падежный окончаний. Если вы назначили ссылку <<< собака, как в предыдущем примере, howt проигнорирует варианты: собаки, собакой, собаку, собаке<sup>41</sup>.

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:

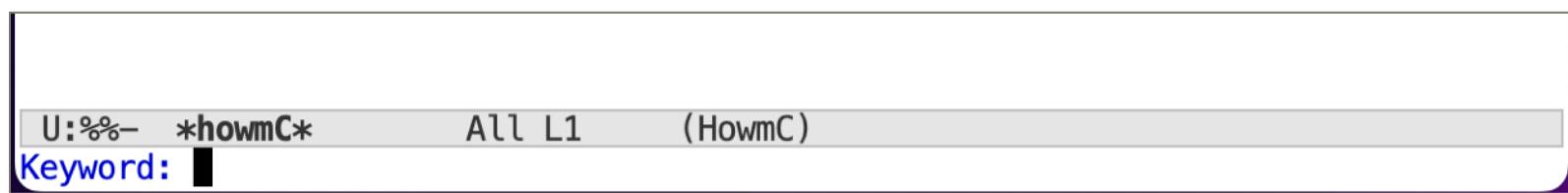
Если вы дополнительно работаете в приложениях, где теги обозначаются через значок решетки, используйте его: <<< #интуиция

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

В Howt есть специальная команда для вызова всех ключевых слов – С-с , і

<sup>41</sup> Ссылки-сюда в howt чувствительны к регистру. Если вы указали слово с большой буквы, то же слово, но с маленькой буквы в других заметках не преобразуется в ссылку. Однако поиск к регистру не чувствителен. Поэтому в поисковой выдаче (промежуточный этап) будут все варианты.

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 сразу<sup>42</sup>.

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 `howm`<sup>43</sup>.

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.

---

<sup>42</sup> Эта и многие другие функции лучше работают, если у вас установлен дополнительный пакет `Vertico`.

<sup>43</sup> См. раздел «Дополнительные настройки», стр. 71.



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 instances of the howm editor (Text Fly) running on macOS. 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 the text, there are two callout boxes with instructions:

- A box with the text: "Создайте имя новой заметки и заключите его в квадратные скобки. Это активирует ссылку." (Create a name for a new note and enclose it in square brackets. This activates the link.)
- A box with the text: "Разместите курсор на ссылке и нажмите RET. Это создаст новую заметку." (Place the cursor on the link and press RET. This creates a new note.)

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
```

Below the windows, a status bar shows:

U:-- 2023-05-30-084133.txt All L6 (Text Fly/-- howm AL Wrap)  
Auto-saving...done

On the left side of the bottom window, there is a box with the text: "Заголовок заметки создан автоматически. Он представляет из себя ссылку-сюда. Так что любые упоминания имени Пушкин теперь будут ссылаться на эту заметку. Имя Пушкин включено в файл .howm-keys". Red arrows from the text in this box point to the title and the link in the bottom window.

On the right side of the bottom window, there is a box with the text: "Ссылка на Заметку А. Howm добавляет такие ссылки всякий раз, когда одна заметка создается из другой." A red arrow points from the text in this box to the link in the bottom window.

Достоинства ссылок в квадратных скобках:

1. Вы создаете ссылку, howm 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.

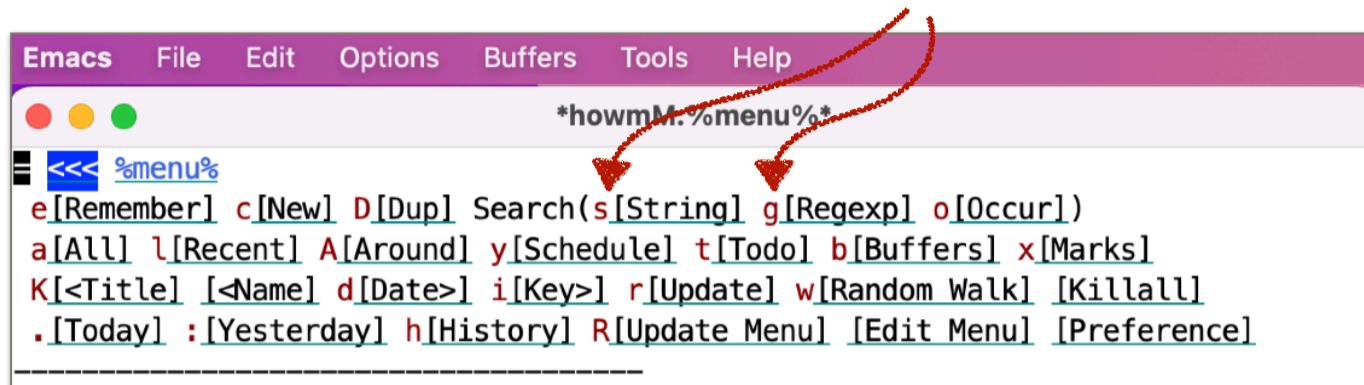
Недостатки подобных ссылок:

1. Подобные ссылки хорошо работают лишь с заголовками заметки. Однако нумерические ссылки надежнее и универсальнее.
2. Заголовки заметок будут скапливаться в файле .howm-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 И ФИЛЬТРАЦИИ

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` нет<sup>44</sup>. 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 автозаполнения<sup>45</sup>.

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».

<sup>44</sup> Вы также можете перейти к поиску сразу после запуска Emacs.

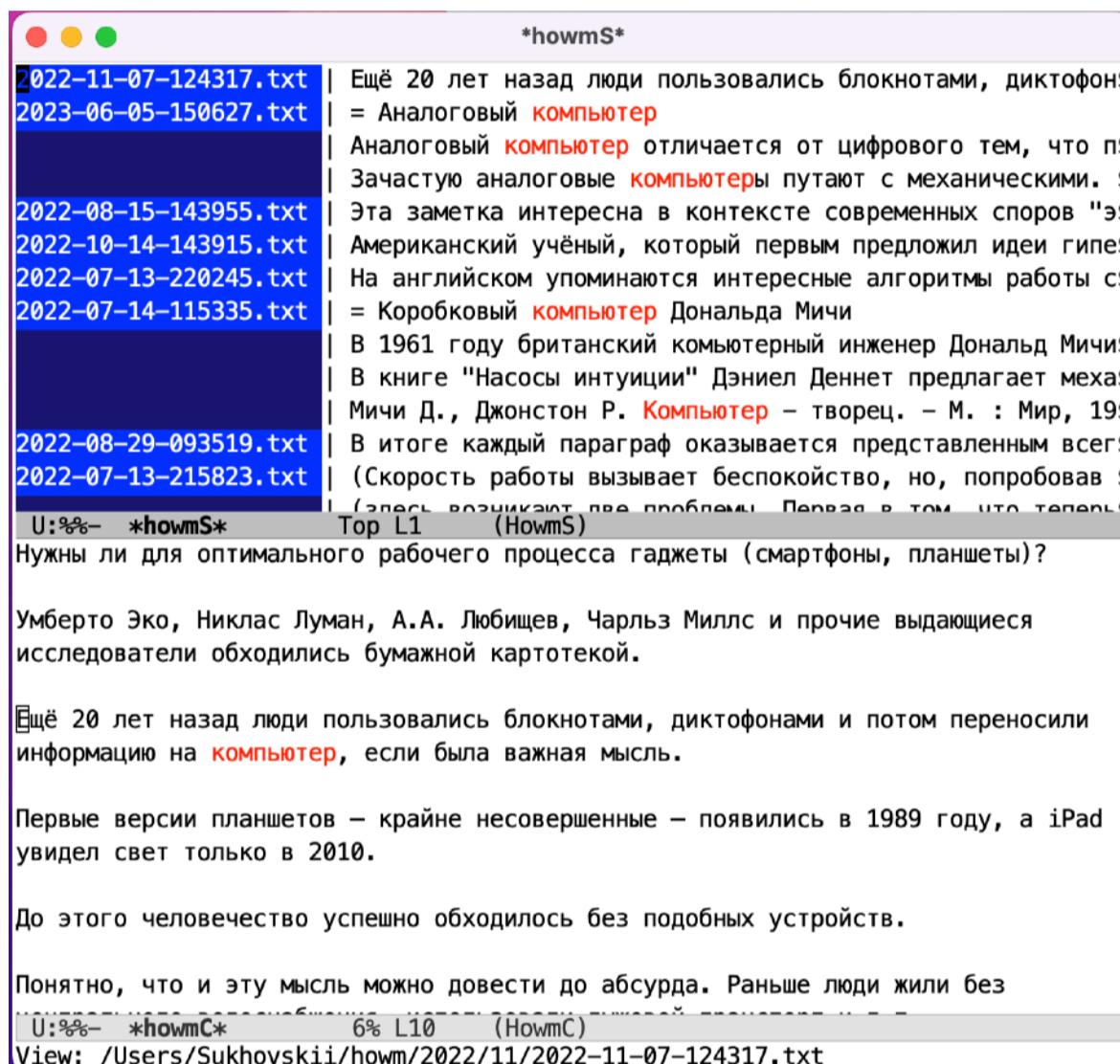
<sup>45</sup> Еще раз отмечу, что с пакетом `Vertico` это делать удобнее. Есть, правда, один побочный эффект. `Vertico` слишком услужлив. Предположим, в список ключевых слов у вас включено понятие «филология». Если вы решите найти записи со словом «филолог», ничего не получится. Буквы в вашем поисковом 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).



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 `i`. В списке останутся выдержки с первым упоминанием слова в заметке. Чтобы вернуть подробную сводку, нажмите на `i` 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 завязана на клавишу `G`. После поиска по первому слову, нажмите `G` и введите второе слово.

Вы можете отсеять заметки из поисковой выдачи и другим способом.

Нажмите клавишу `f`. `Howt` 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.

The screenshot shows a terminal window with two buffers. The top buffer, titled 'U:%%- \*howtC\*', displays a completion menu with the message 'Click on a completion to select it.' and 'In this buffer, type RET to select the completion near point.' Below this, under 'Possible completions are:', there is a grid of words:

Around	Region	contents	date
mtime	name	reminder	summary
uniq			

The bottom buffer, titled 'U:%%- \*Completions\*', shows a list starting with 'filter by:'. The entire window has a light gray background and a dark gray header bar.

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**. Nowt 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**

Nowt 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.



Пожалуй, стоит пояснить, зачем все эти ухищрения нужны. Для этого следует рассмотреть, какое место занимают списки в рабочем процессе с howm.

Представьте себе бумажную картотеку. Вы достаете стопку карточек из ящика и раскладываете их на столе. Какие-то вам пригодятся для написания статьи/подготовки к выступлению, какие-то лишние. Возможно, вы используете ключевые слова или еще какие-то метаданные, чтобы разгрести завал. На столе должна остаться лишь коллекция карточек, необходимых для работы.

Список в howm может быть таким «столом» с разложенными карточками.

К сожалению, в howm 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.

В howm, тем не менее, есть простая история поиска, которая вызывается командой - C-c , h

# УДАЛЕНИЕ ЗАМЕТОК

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

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.

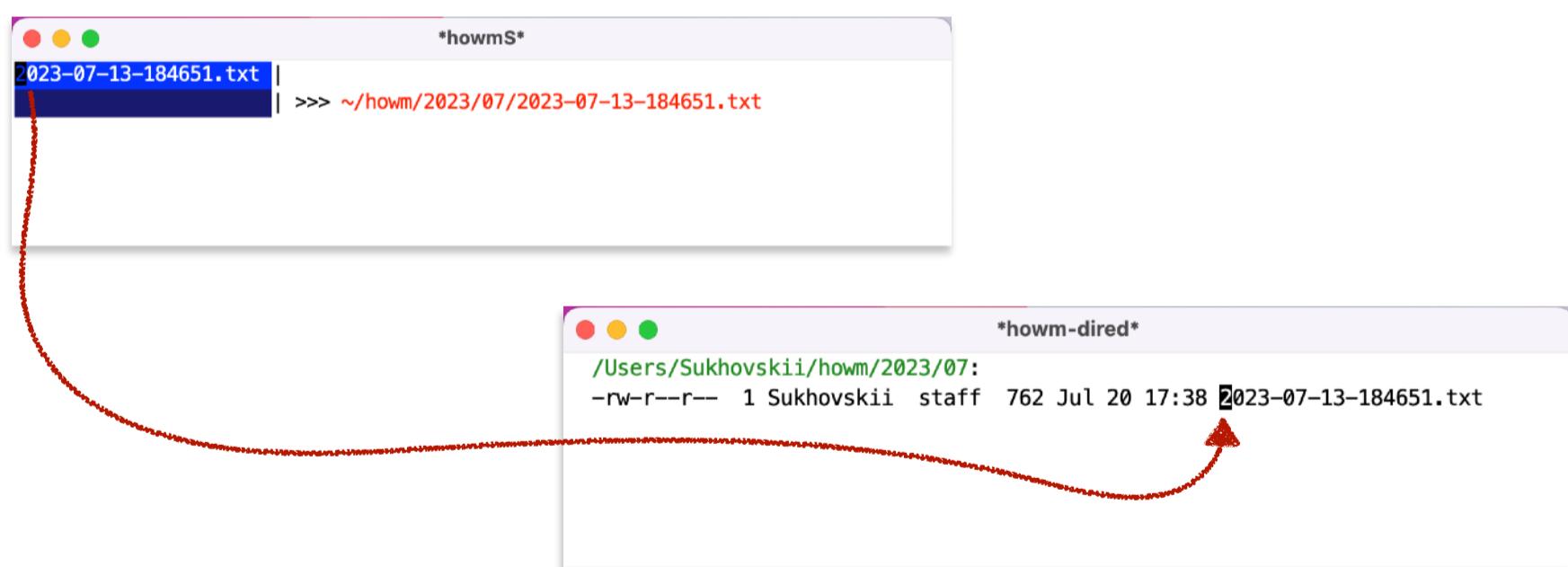
Самый простой способ проверки таков:

Вставьте в заметку ссылку на нее саму<sup>46</sup>. Переместите курсор на ссылку и нажмите RET. В списке должна оказаться всего одна заметка.

Если вместо нумерических ссылок вы используете ссылки на заголовки, процедура почти идентична. Копирование заголовков осуществляется командой С-с , К

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.



В новом окне вам доступны все стандартные для режима `Dired` манипуляции с файлами<sup>47</sup>.

<sup>46</sup> Мы уже обсуждали, как это сделать на примере обратных ссылок. См. стр. 31

<sup>47</sup> Как здесь, так и в других случаях, знакомство с документацией `Emacs` существенно расширит свободу ваших действий.

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 файл<sup>48</sup>.

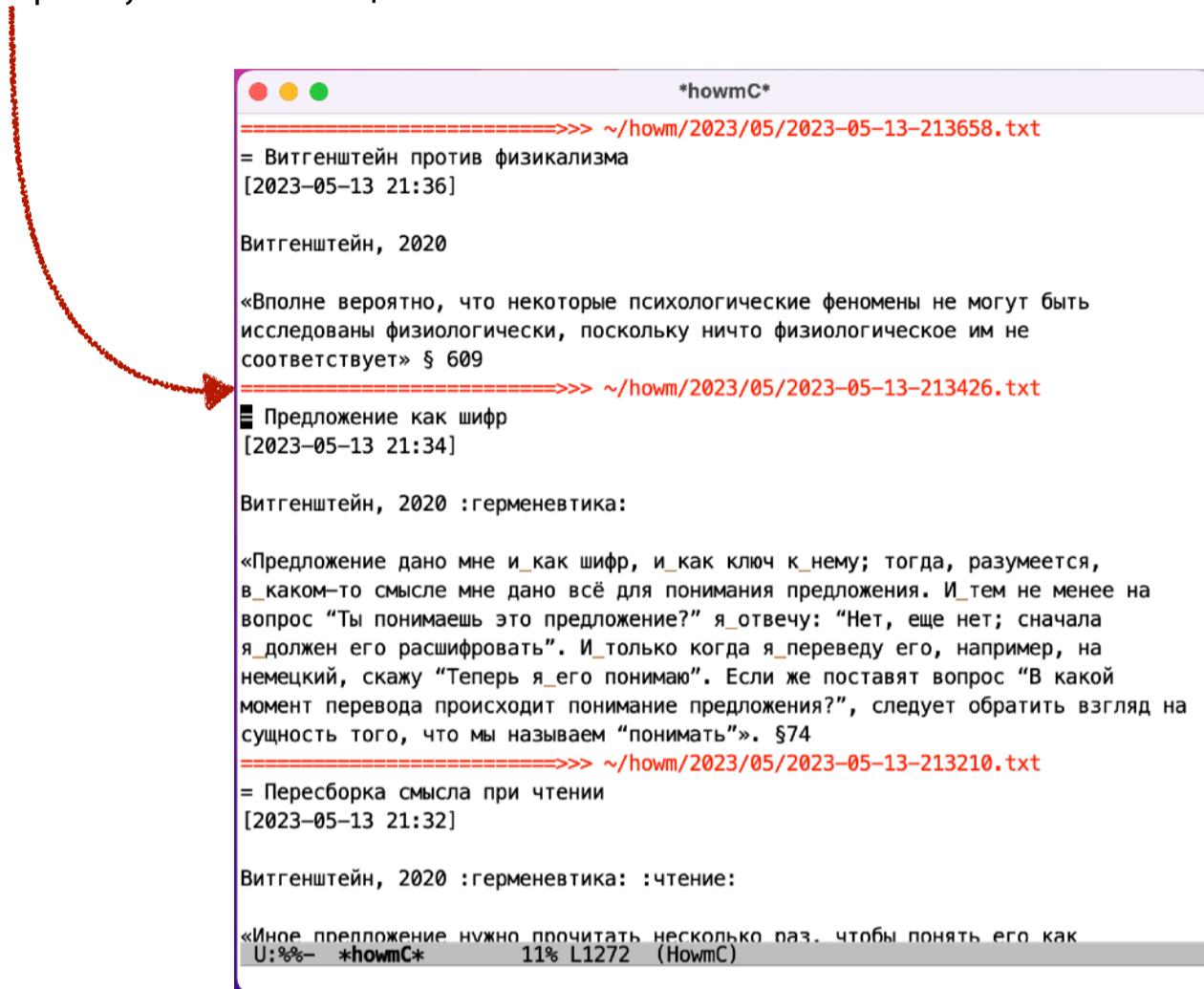
Вы можете выбрать сразу несколько файлов для удаления. Для этого нажимайте последовательно клавишу d. Emacs пометит файлы. Для подтверждения операции следует нажать x. Если вы решили снять метку с какого-либо файла, переместите курсор нужную строку и нажмите u.

---

<sup>48</sup> 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 заметок<sup>49</sup>. 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: С-с , s
2. Объедините результат в единый свиток.
3. Воспользуйтесь поиском по тексту, чтобы найти нужное место: С-с

<sup>49</sup> 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 редактор<sup>50</sup> 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 автозамены по файлам.

---

<sup>50</sup> 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 автозамену. Нажмите клавишу `Y`, чтобы сделать это сразу во всех файлах (метка уникальна и 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.

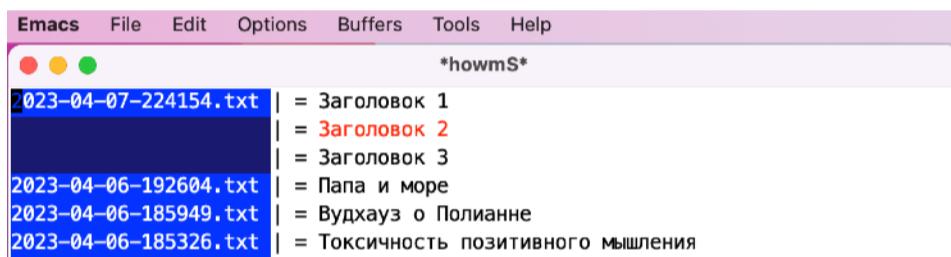
# РАБОТА С ДЛИННЫМИ ЗАМЕТКАМИ

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:

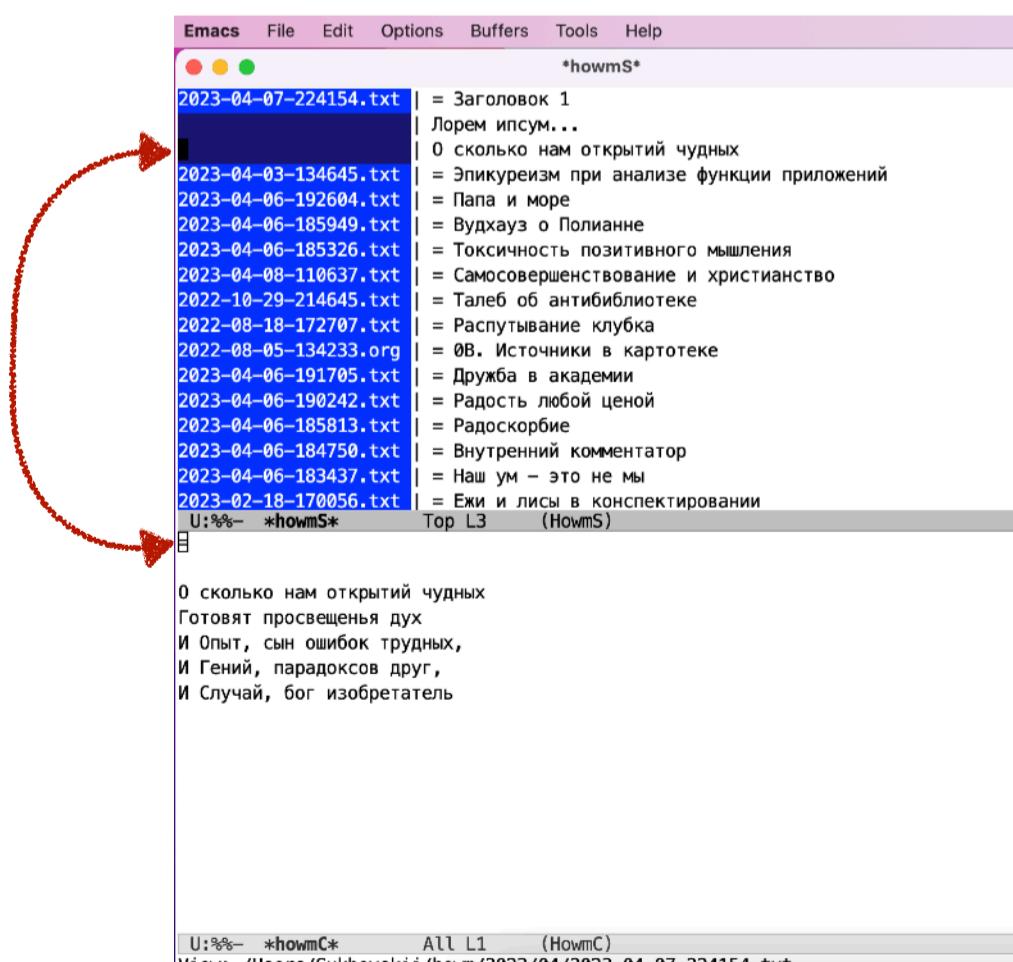
```
= заголовок 1  
= заголовок 2  
= заголовок 3
```

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.

Заметьте, 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 заметка<sup>51</sup>.



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.

<sup>51</sup> 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)<sup>52</sup>.

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. Свернуть 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

Еще один совет.

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?

<sup>52</sup> 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 , }

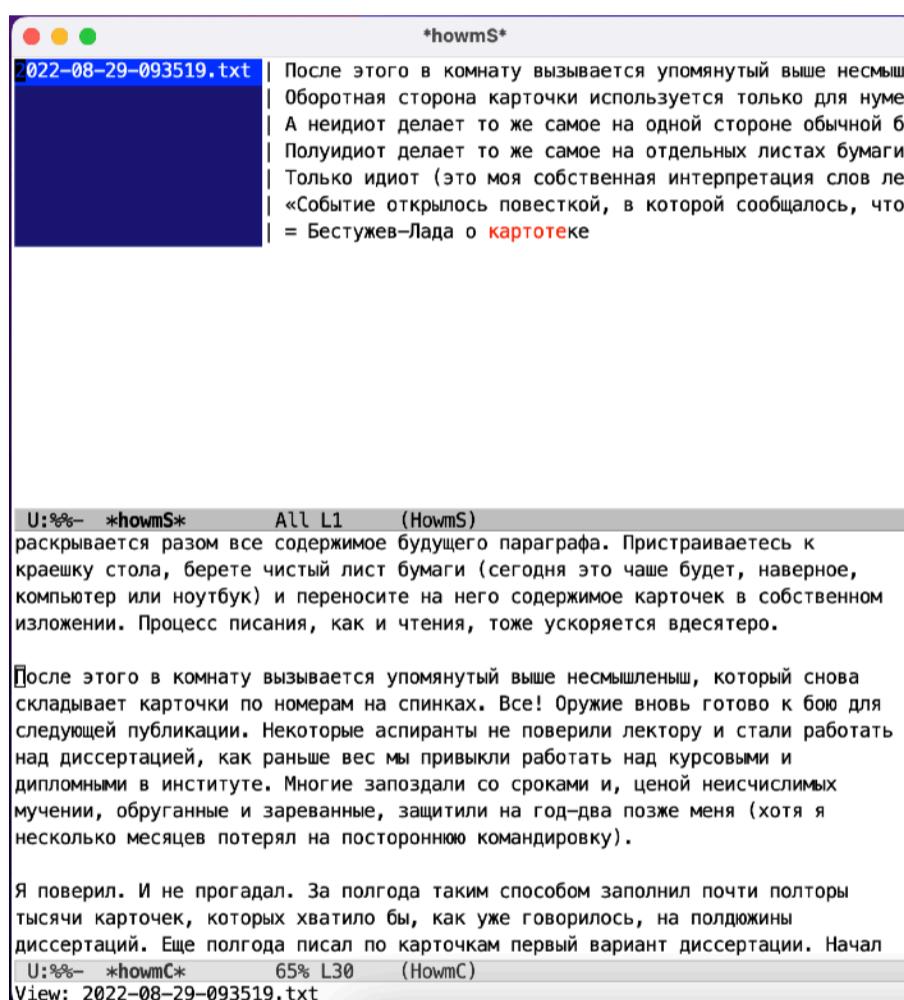
Находясь в заметке, наберите команду `C-c , o`, и вы получите стандартный для `howm` интерфейс: вверху список, внизу текст. Только в данном случае, вы будете перемещаться в рамках одной заметки.

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 пролистывания<sup>53</sup>.



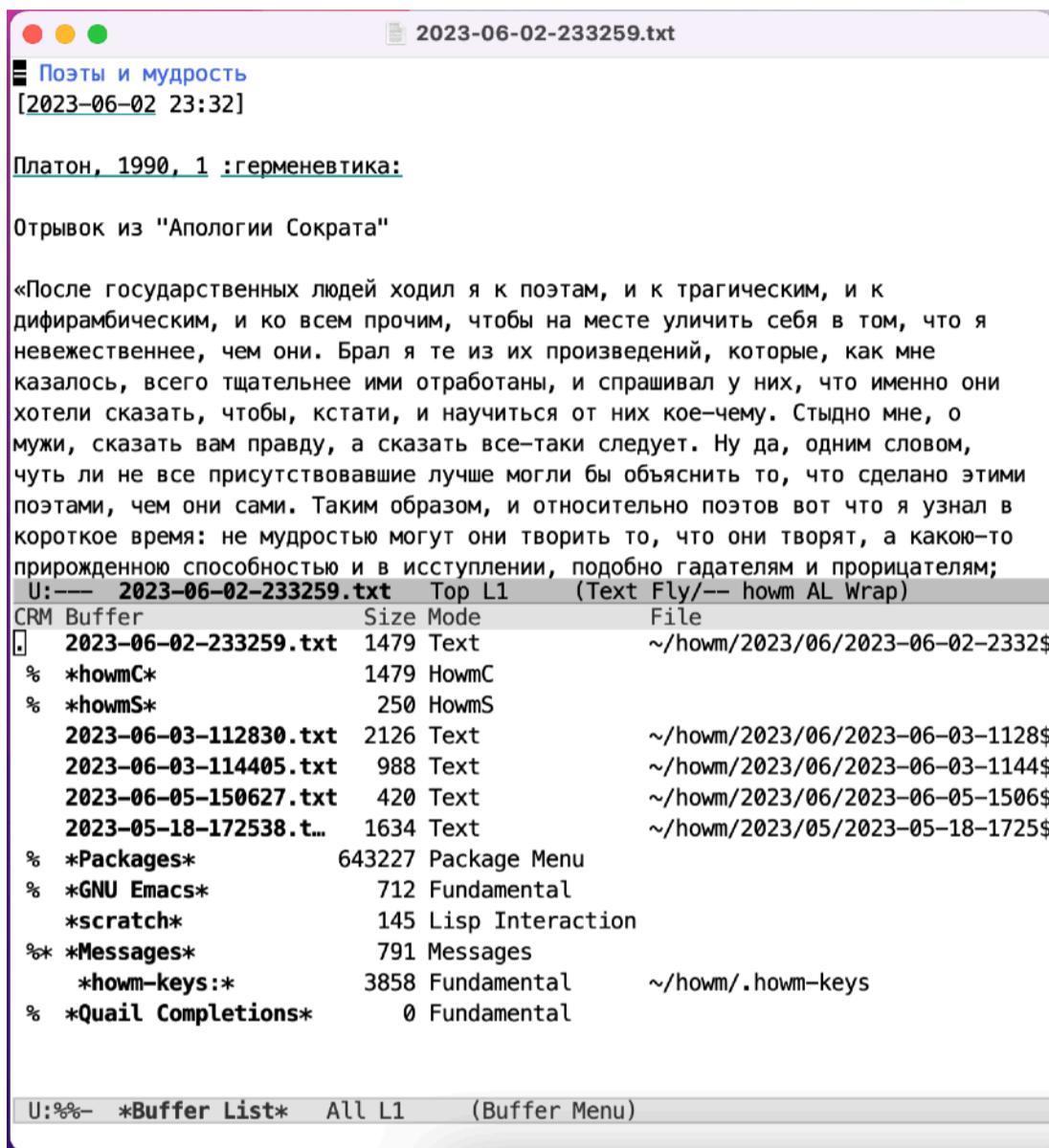
Поскольку ссылка в `howm` – это триггер поискового запроса, вы можете использовать специальные метки, чтобы сослаться на конкретный отрывок в длинном тексте. Например, в одной заметке вы разместите метку – `qq`, а в другой ссылку `>> qq`<sup>54</sup>. При переходе по ссылке `howm` откроет документ в нужном месте.

<sup>53</sup> 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.

<sup>54</sup> 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 Emacs<sup>55</sup>.

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

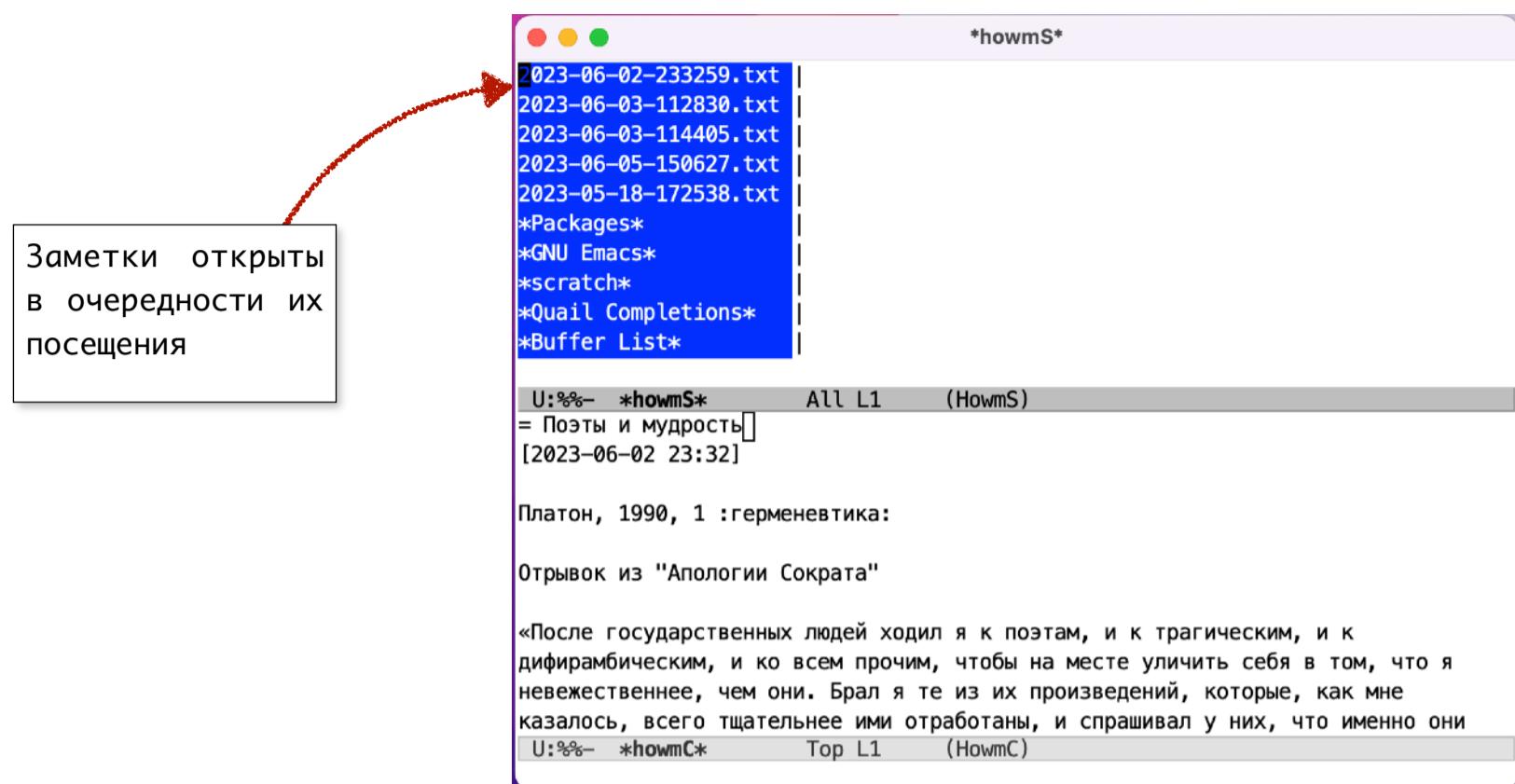
Вас перебросит в последний документ, открытый не в howm-режиме. Набрав команду еще раз, вы вернетесь обратно в заметку<sup>56</sup>.

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.

<sup>55</sup> См.: [https://www.gnu.org/software/emacs/manual/html\\_node/emacs/Buffers.html](https://www.gnu.org/software/emacs/manual/html_node/emacs/Buffers.html)

<sup>56</sup> 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. 74). 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 сессии<sup>57</sup>.

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`<sup>58</sup>.

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

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

<sup>57</sup> В списке отображаются нумерические имена файлов. Заголовки были бы понятнее. К счастью, можно внести соответствующие усовершенствования. См. «Дополнительные настройки», стр. 73

<sup>58</sup> Стандартная команда закрытия буфера – `C-x k` тоже работает.

# БЫСТРЫЕ ЗАМЕТКИ

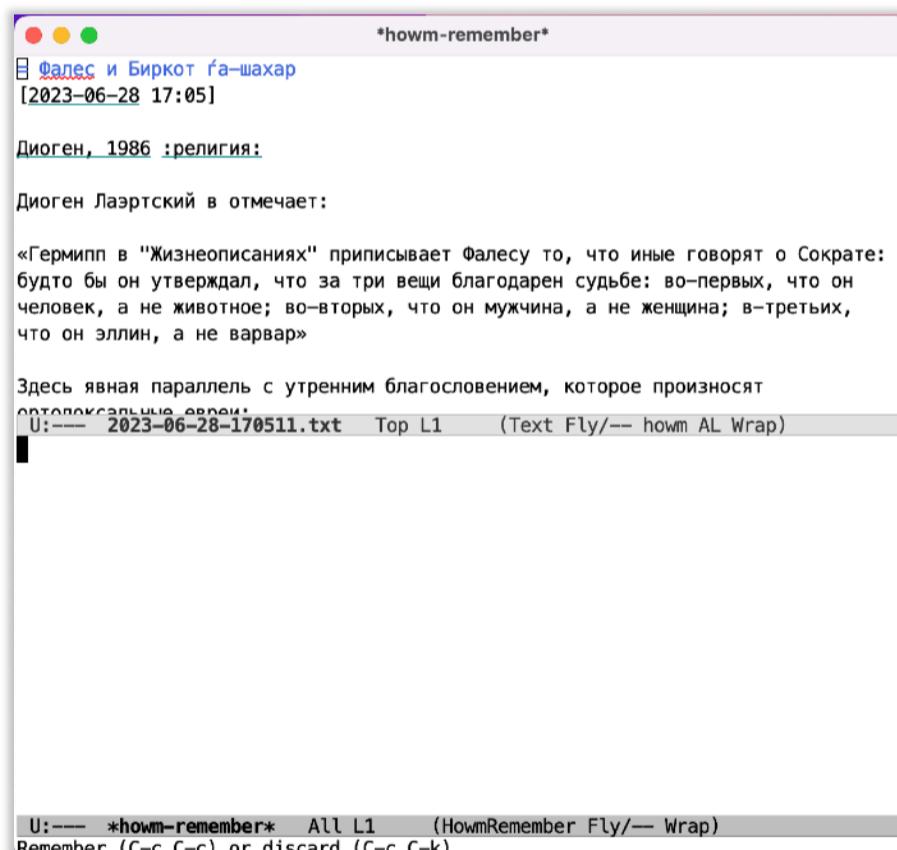
Порой вы пишите одну заметку, а на ум приходит идея для другой. Вы не хотите прерывать состояние потока, но также не хотите упустить инсайт.

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.



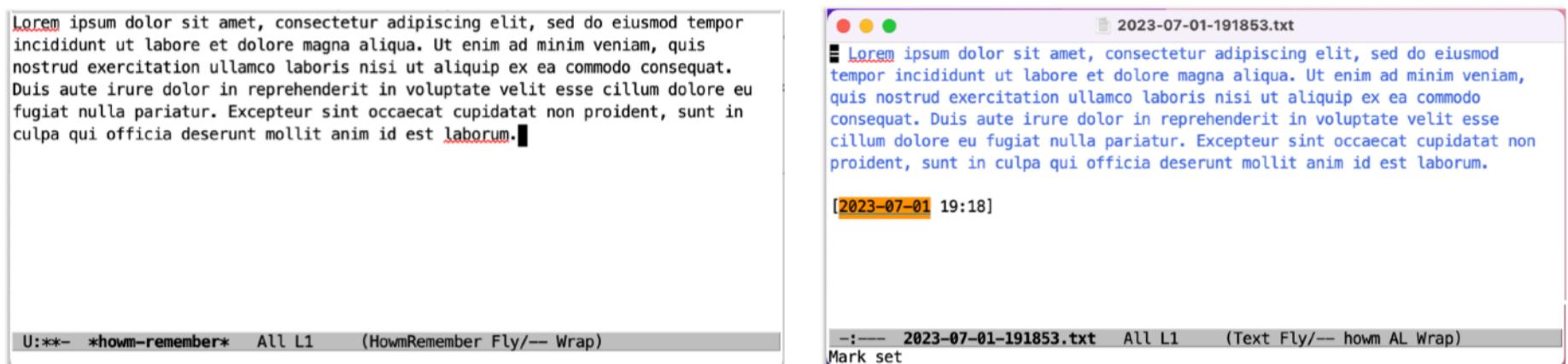
Идеи, попавшие сюда, еще не получили прописки в картотеке. Если вы решите, что они неуместны, нажмите `C-c C-k` и `howm` закроет (убьет) буфер, ничего не сохраняя<sup>59</sup>.

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.

<sup>59</sup> 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.

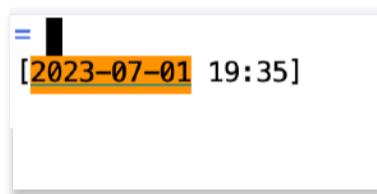


Конечно, вы можете переинициализировать эту функцию для создания развернутых черновиков. Однако сделать это можно будет только путем глобального изменения настроек. Внесите в файл .emacs следующую строчку:

```
(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 одинаковыми<sup>60</sup>.

<sup>60</sup> Подробности в разделе «Дополнительные настройки», стр. 70-71

# 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 `C-c , 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. 51) 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 и закройте окно. Никаких следов в системе не останется<sup>61</sup>.

<sup>61</sup> Тут все также, как и отменой при создании заметки (см. стр. 18)

# НЕФОРМАТНЫЕ ЗАМЕТКИ

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 ~/home/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`

# УПРАВЛЕНИЕ ДЕЛАМИ 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»<sup>62</sup>. 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)<sup>63</sup>.

Хираока также 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.

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.

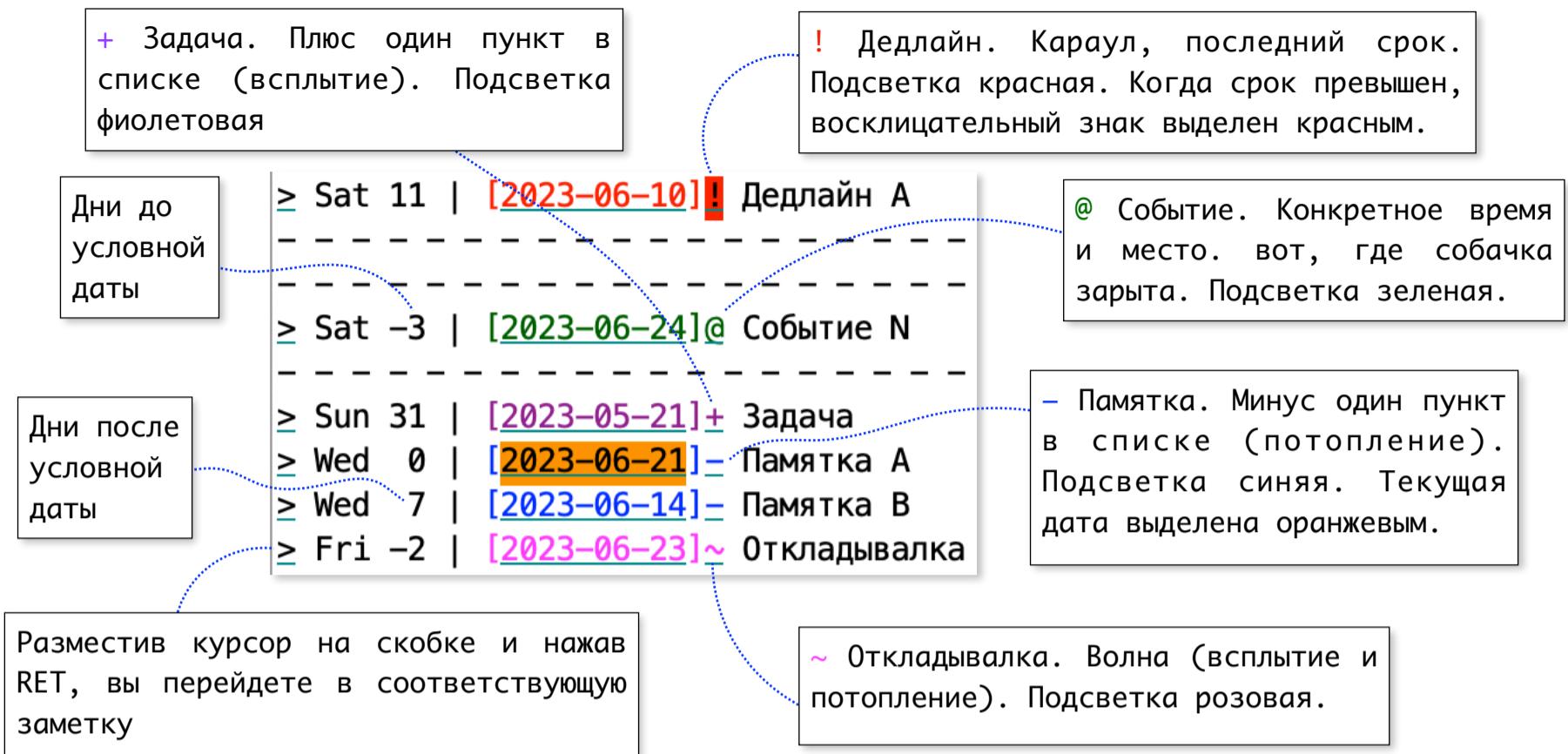
Для обозначения разного типа планов в howm существует особый синтаксис, с мнемотехническими подсказками.

---

<sup>62</sup> Схожие идеи озвучил Дэвид Ален в работе «Как привести дела в порядок».

<sup>63</sup> Чуть ниже мы рассмотрим, как это все это выглядит в howm.

## Синтаксис планировщика



**Событие** появляется в списке за семь дней до назначенной даты и мгновенно всплывает наверх. Просроченные события убираются из списка сразу на следующий день. Вполне разумно, так как вам не нужно напоминание о концерте, который состоялся вчера.

**Дедлайн** находится внизу списка. За семь дней до даты поднимается наверх и остается на плаву даже по истечении срока.

**Задача** находится внизу списка. Начинает постепенно всплывать наверх списка за семь дней до назначенной даты.

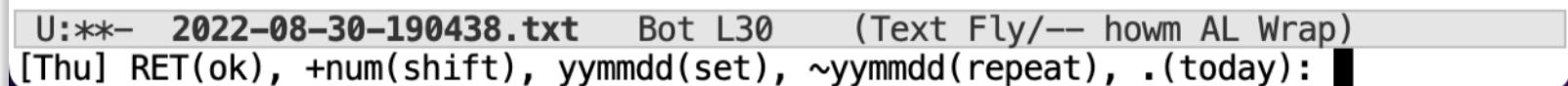
**Памятка** находится внизу списка. Оказывается наверху в назначенную дату и затем тонет понемногу каждый день, погружаясь вниз списка<sup>64</sup>.

**Откладывалка** ведет себя как памятка. Отличие в том, что всплытия и погружения цикличны. Откладывалка будет всплывать снова и снова с периодичностью в 30 дней.

Какой бы тип плана ни был, в howt он будет привязан к дате. Формат даты: [YYYY-MM-DD]. Можно внести эти данные вручную, однако в howt существуют и более замысловатые способы.

<sup>64</sup> Количество пунктов погружения будет обусловлено датами других задач и памяток.

Команда `C-c , d` вставит нужную дату в открытой заметке<sup>65</sup>. Для этого howt откроет диалог в минибуфере:



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

Функцию `repeat` мы рассмотрим чуть дальше. Сейчас самое простое. Если нужна сегодняшняя дата, 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. Howt может сообразить, о чём идет речь.

Цифра от 1 до 31 будет истолкована как число текущего месяца<sup>66</sup>. 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 наберете `{_}` и, разместив курсор внутри, нажмете RET, эффект будет идентичным<sup>67</sup>.

Так или иначе теперь вы мастер точных датировок. Что дальше? После даты нужно указать тип плана и пояснение. Например:

`[2023-12-30]! Заполнить годовой отчет.`

Вы можете создать отдельную заметку с планами или размещать их в разных заметках. В одной заметке вы можете перечислять задачи подряд или перемежать с комментариями. Это не имеет значения. Howt позволяет быстро ориентироваться в любом многообразии задач. Если вы сохраните файл с задачей и перейдете в основное меню, то увидите, что она включена в сводный список и скоординирована с другими планами.

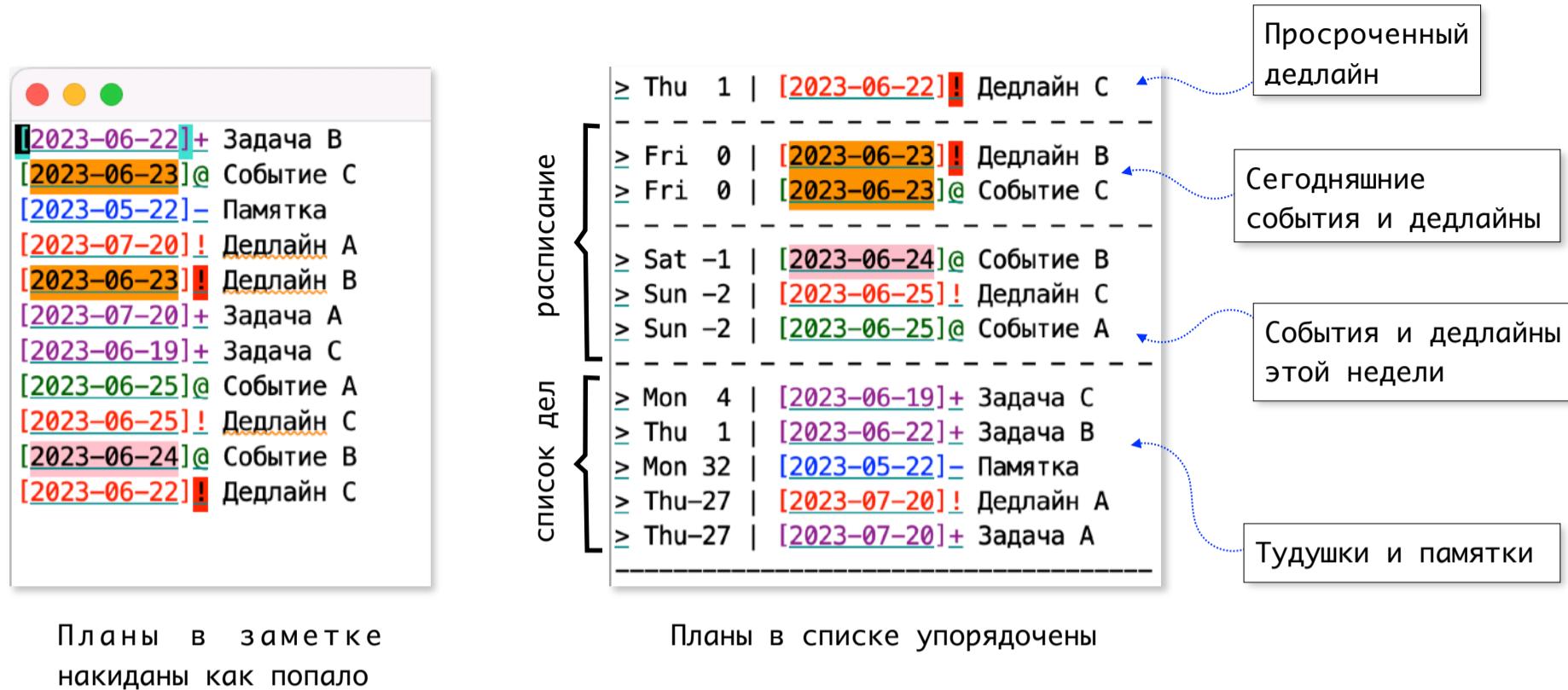
---

<sup>65</sup> В меню howt есть клавиша `d[Date>]`, которая вставляет дату в последнюю открытую заметку (последний буфер). Однако, если вы еще открыли никаких заметок, команда не сработает. Плюс, дата вставится туда, где располагался курсор.

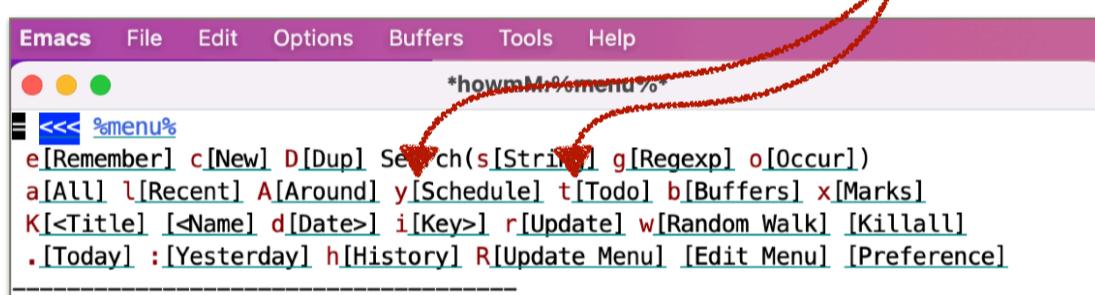
<sup>66</sup> Если на дворе июнь, а вы набрали цифру 31, howt истолкует это как 30+1 и решит, что речь про первое июля.

<sup>67</sup> Это один из примеров функции, которая называется `action-lock`. С её помощью в howt можно автоматизировать множество процессов (см. «Дополнительные настройки», стр. 74-75)

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 расписания<sup>68</sup>.



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 **C-c , t**

Вы получите интерфейс, привычный по работе с заметками.

Преимущество (особенно, если задачи рассредоточены по разным файлам) в том, что вы можете осуществить быстрый предпросмотр запланированного.

Список запланированных событий вызывается командой – **C-c , u**

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

В отличие от основного меню `howm` списки не лимитированы количеством пунктов. Вы получаете доступ ко всем планам<sup>69</sup>.

<sup>68</sup> Расписание показывает события и дедлайны в порядке намеченных дат. С другой стороны, памятки и задачи отображаются в «сезонном порядке».

<sup>69</sup> Для примера, события, назначенные на следующий квартал, из основного меню видны не будут.

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`. Howm перестроит список в согласии с датами.



Вы можете интегрировать планы, размещенные в `howm`, 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`. `Howm` откроет диалог, который мы уже рассматривали на стр. 63.

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 `howm`. Действия те же. Переместите курсор внутрь квадратных скобок с датой и нажмите `RET`. `Howm` временно перебросит вас в нужный файл с открытым диалогом в минибуфере. После выполнения операции вы автоматически вернетесь в меню<sup>70</sup>.

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

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`.

<sup>70</sup> Учтите, что изменения пока не сохранены. Для того, чтобы сделать это, не покидая меню, используйте команду `C-x S`

Так же как и в случае с датами вы можете вносить изменения в записи прямо из главного меню `howm`<sup>71</sup>.

Кроме изменения типа мероприятия вы можете задать параметры его поведения (быстроты всплытия/потопления в списке). Для этого после символа (`~@-!+`) нужно указать цифру. В минибуфере, который мы рассмотрели на предыдущей странице, есть опция: `num (laziness)`. Вас достаточно указать нужную цифру и нажать RET. Чем выше число, тем медленнее движение вниз и быстрее движение вверх.

Разберем на примерах.

По умолчанию событие пропадает с радаров на следующий день после запланированной даты. Указав `[2023-06-25]@5` вы сделаете так, что событие будет на плаву еще пять дней.

Дедлайн выныривает на поверхность списка за 7 дней до даты. Если вы хотите увидеть предупреждение о дедлайне за 20 дней, наберите `[2023-06-25]!20`

Памятка всплывает наверх к назначеннй дате, потом тонет. Скорость затопления регулируется. Запись `[2023-06-25]-9` будет погружаться каждые девять дней. Памятка с суффиксом `-999` останется на плаву годами.

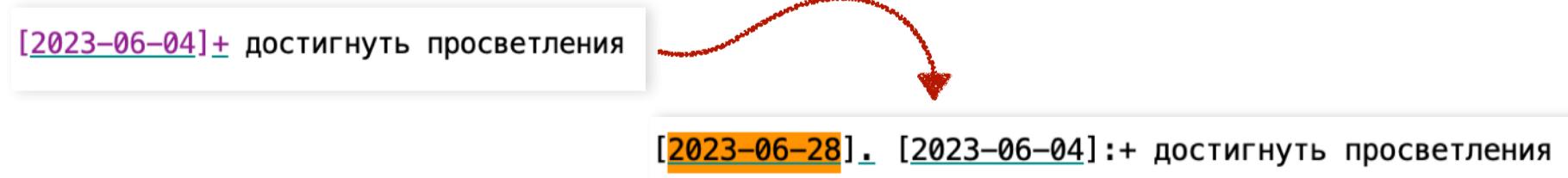
Задача начинает постепенно подниматься на поверхность списка за 7 дней до даты. Если вы хотите чтобы задача начала всплывать раньше, увеличьте число: `[2023-06-25]+14`

В случае с откладывалкой число будет означать длину цикла потопления-выныривания. Если вы хотите видеть запись наверху списка чаще одного раза в 30 дней, укажите это: `[2023-06-25]~15`

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

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 изменится.



<sup>71</sup> Конечно, если вы редактируете планы, находясь в самой в заметке, достаточно просто стереть символ и впечатать новый (количество кликов одинаковое).

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

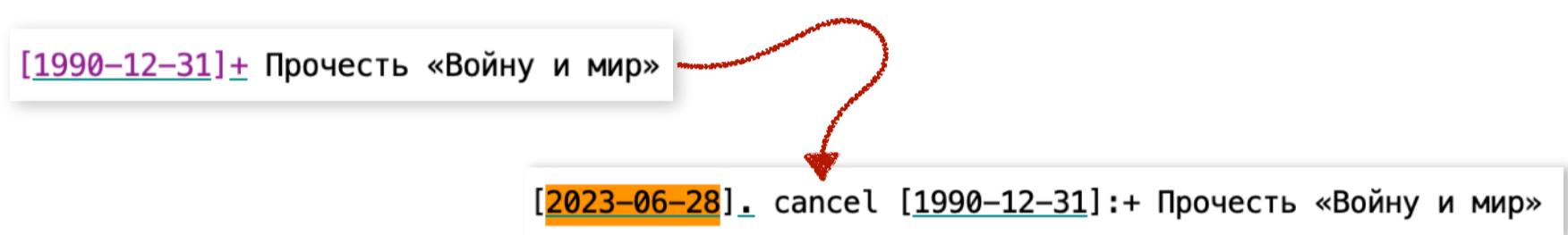
Однако вы можете использовать и упрощенный синтаксис. Разместите где-нибудь примечание:

[2023-06-28]. достигнуть просветления

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

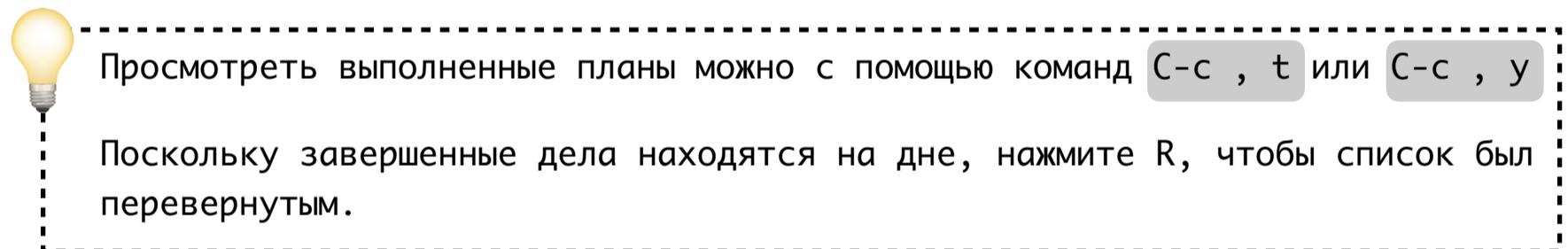
Другой возможный исход – отмена плана. За всем не угнаться, ничего страшного.

Находясь в диалоге минибуфера, нажмите клавишу X



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<sup>72</sup>.



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

Howt 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<sup>73</sup>. 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).

<sup>72</sup> В главном меню их и вовсе нет.

<sup>73</sup> Заметьте, однако, что откладывалки циклически.

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:***- 2022-08-30-190438.txt Bot L30 (Text Fly/-- howm AL Wrap)
[Thu] RET(ok), +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)] █
```

Еще одно нажатие на RET – ежедневно до указанной даты.

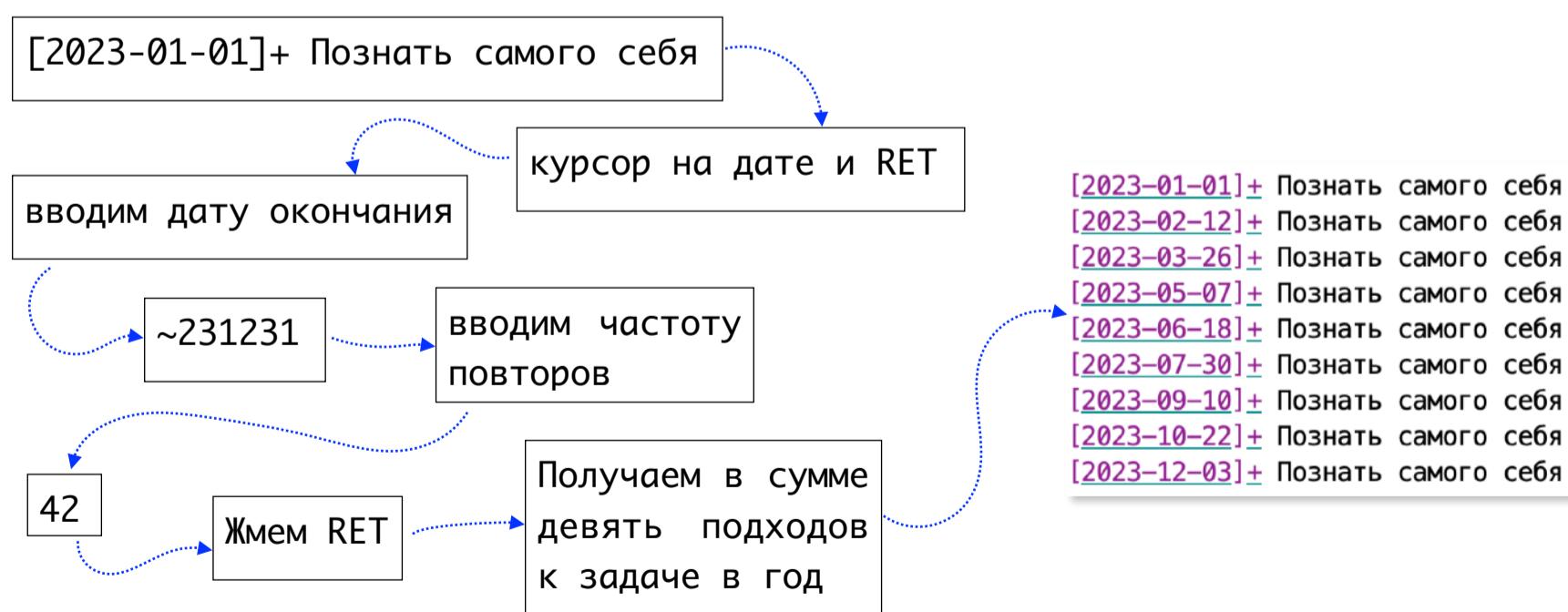
Любая цифра – количество дней между повторами.

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.

Пример:



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).

# ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ

Основной способ настройки howm: M-x customize → Applications → Howm.

Если какой-то элемент системы вас не устраивает, выберите альтернативу из списка с помощью кнопки Show.

This is GNU Emacs 28.2 (build 1, x86\_64-apple-darwin18.7.0, NS appkit-1671.60 Version 10.14.6  
(Build 18G95))  
of 2022-09-12  
U:\*\*\*- \*GNU Emacs\* All L5 (Fundamental)  
M-x customize

1. Перейдите в минибуфер командой M-x
2. Наберите customize
3. Нажмите RET

\*Customize Group: Emacs\*

For help using this buffer, see [Easy Customization](#) in the [Emacs manual](#).

Search

Operate on all settings in this buffer:

Revert... Apply Apply and Save

**Emacs group:** Customization of the One True Editor.

**State:** visible group members are all at standard values.  
See also [Manual](#).

[Editing](#) Basic text editing facilities.  
[Convenience](#) Convenience features for faster editing.  
[Files](#) Support for editing files.  
[Wp](#) Support for editing text files. [More](#)  
[Text](#) Support for editing text files.  
[Data](#) Support for editing binary data files.  
[External](#) Interfacing to external utilities.  
[Communication](#) Communications, networking, and remote access to files.  
[Programming](#) Support for programming in other languages.  
[Applications](#) Applications written in Emacs.

В открывшемся списке найдите пункт Application.  
Наведите курсор и нажмите RET.

\*Customize Group: Applications\*

For help using this buffer, see [Easy Customization](#) in the [Emacs manual](#).

Search

Operate on all settings in this buffer:

Revert... Apply Apply and Save

Parent groups: [Emacs](#)

**Applications group:** Applications written in Emacs.

**State:** visible group members are all at standard values.

**Subgroups:**

[Calc](#) Advanced desk calculator and mathematical tool.  
[Calculator](#) Simple Emacs calculator.  
[Calendar](#) Calendar and time management support.  
[Doc View](#) In-buffer viewer for PDF, PostScript, DVI, and DJVU files.  
[Erc](#) Emacs Internet Relay Chat client.  
[Eshell](#) Command shell implemented entirely in Emacs Lisp. [More](#)  
[Games](#) Games, jokes and amusements.  
[Howm](#) Wiki-like note-taking tool.  
[Htmlfontify](#) Convert buffers and files to HTML.

В новом списке найдите пункт Howm.  
Наведите курсор и нажмите RET.

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 Revert.  
Можно просто удалить соответствующие строки из custom-set-variables. Howm забудет о всех модификациях.

\*Customize Group: Howm\*

For help using this buffer, see [Easy Customization](#) in the [Emacs manual](#).

Search

Operate on all settings in this buffer:

Revert... Apply Apply and Save

Parent groups: [Applications](#)

**Howm group:** Wiki-like note-taking tool.

**State:** visible group members are all at standard values.

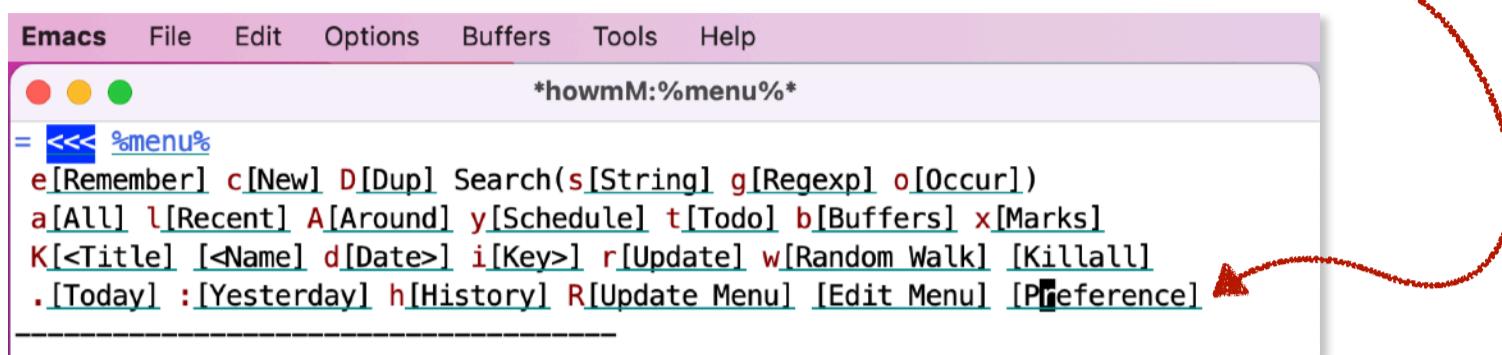
**Subgroups:**

[Howm Compatibility](#) Compatibility to howm-1.3.\*.  
[Howm Devel](#) Developers' diagnoses.  
[Howm Efficiency](#) To improve performance, use grep and turn off expensive options.  
[Howm Experimental](#) Test of experimental features.  
[Howm Faces](#) Colors and fonts.  
[Howm Files](#) Names of files and directories.  
[Howm Hooks](#) Hooks.  
[Howm List](#) Style of list view.  
[Howm Menu](#) Menu.  
[Howm Misc](#) Miscellaneous customization.  
[Howm Reminder](#) Schedule and todo list.  
[Howm Search](#) Search methods.

U:\*\*\*- \*Customize Group: Howm\* Top L2 (Custom)

На самом деле я продемонстрировал вам не самый оптимальный маршрут к настройкам (он хорошо тем, что универсален и подходит к другим приложениям).

Вы можете перейти к настройкам `howm` из основного меню.

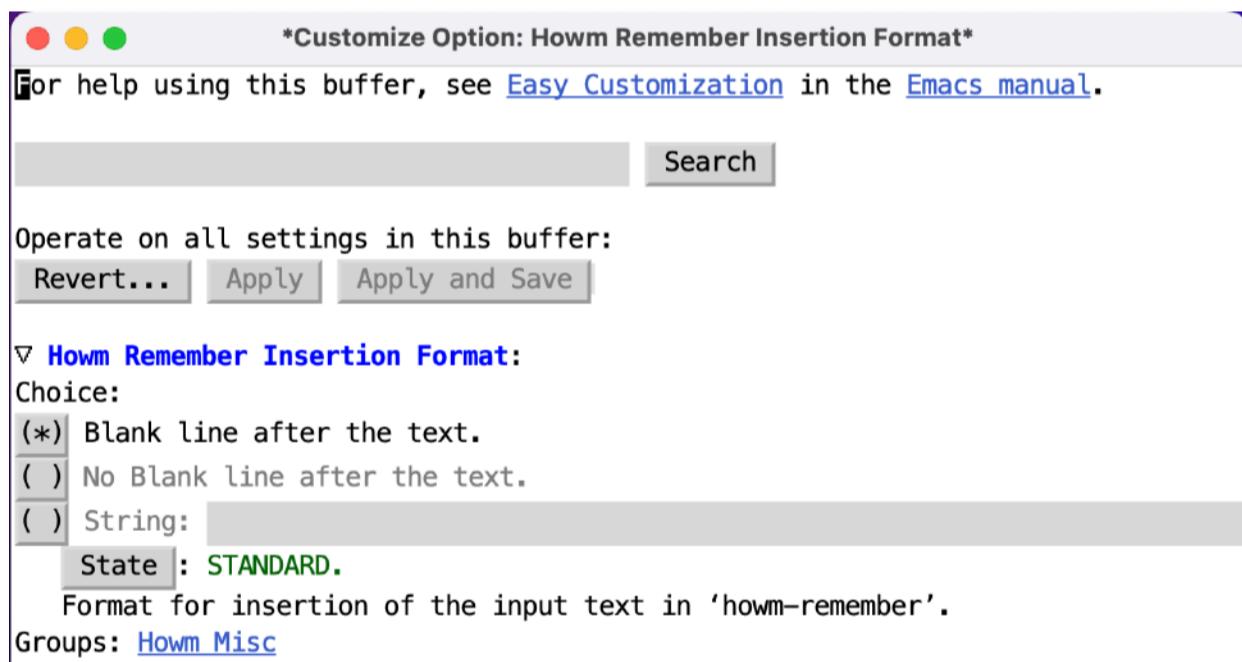


Кроме того, множество настроек можно внести в файл `.emacs` собственноручно. Разумеется, делать это стоит с осмотрительностью.

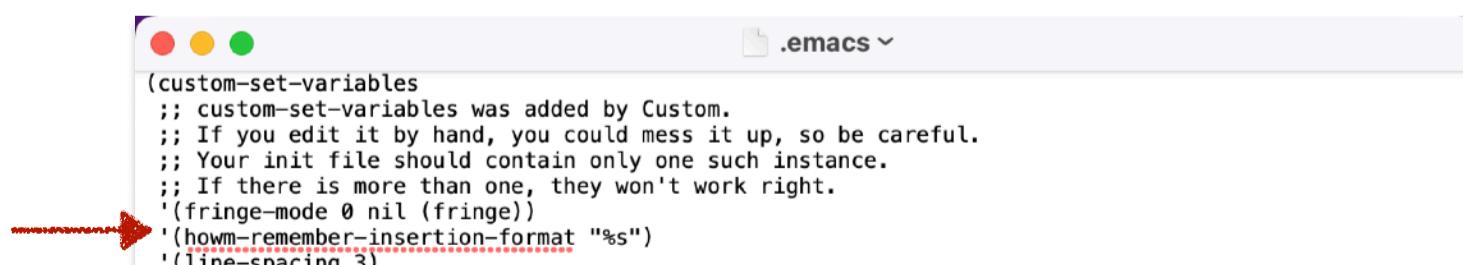
Рассмотрим конкретный пример. Вы хотите убрать пустую строку в шаблоне быстрых заметок. Вызовите меню с настройками:

M-x customize-viriable RET `howm-remember-insertion-format` RET

Пред вами откроется диалог. Выберите вторую опцию: No Blank line...



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



Если вы решите изменить настройки, Emacs автоматически ее перепишет.

Вы можете и сами записать в файле `.emacs` соответствующую команду:

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

Опытные пользователи часто прибегают к прямой редактуре. Однако следует иметь в виду, что отмотать такие изменения назад через настройки уже не получится. Нужно будет опять залезать в `.emacs` для коррекции<sup>74</sup>.

. . .

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")
```

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

---

<sup>74</sup> Если вы вносите одни изменения через панель настроек, а другие напрямую, возможен конфликт. Желательно вести для себя протокол изменений, чтобы не запутаться.



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 заметок<sup>75</sup>. Вернуться к режиму предпросмотра можно нажав клавишу 2, пробел или v



Когда количество заметок переваливает за пару сотен, `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<sup>76</sup>.

---

<sup>75</sup> Без этой настройки при перемещении курсора по списку `Howm` будет снова инициировать предпросмотр.

<sup>76</sup> Есть побочные эффекты! Меню перестанет обновляться автоматически. Но это не страшно. В самом меню есть команда обновления содержания - R[Update Menu]. Другой подобный 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 приплюсовывать новые записи в уже созданную заметку, снабжая из заголовками с датой.

Работать это будет так, как описано на стр. 51-54, 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 включен<sup>77</sup>. 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 потребует 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.

---

<sup>77</sup> 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 элемент<sup>78</sup>.

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:

>>> Заметка

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.

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



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

<sup>78</sup> Не забудьте сделать резервную копию этого файла перед смелыми экспериментами.

# 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 список<sup>79</sup>. 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. 72).

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.

---

<sup>79</sup> 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.

Аккорд	Команда	Действие
C-c , ,	M-x <code>howm-menu</code>	Вызов основного меню
C-c , c	M-x <code>howm-create</code>	Создание новой заметки
C-c , e	M-x <code>howm-remember</code>	Создание быстрой заметки
C-c , I	M-x <code>howm-create-interactively</code>	Создание заметки с произвольным именем и местоположением
C-c , D	M-x <code>howm-dup</code>	Дублирование заметки
C-c , a	M-x <code>howm-list-all</code>	Вызов списка всех заметок
C-c , l	M-x <code>howm-list-recent</code>	Вызов списка недавних заметок (за 7 дней)
C-c , .	M-x <code>howm-find-today</code>	Список сегодняшних заметок
C-c , :	M-x <code>howm-find-yesterday</code>	Список вчерашних заметок
C-c , A	M-x <code>howm-list-around</code>	Записи до и после текущей заметки из списка
C-c , K	M-x <code>howm-keyword-to-kill-ring</code>	Копирование заголовка заметки
C-u C-c , K		Копирование нумерического имени файла вместе с абсолютным путем
C-c , s	M-x <code>howm-list-grep-fixed</code>	Поиск и поиск по ключевым словам
C-c , g	M-x <code>howm-list-grep</code>	Поиск в том числе с <code>regexp</code>
C-c , o	M-x <code>howm-occur</code>	Поиск в теле заметки
C-c , h	M-x <code>howm-history</code>	История поиска
C-c , i	M-x <code>howm-insert-keyword</code>	Вызов ключевых слов из заметки
C-c , p	M-x <code>action-lock-goto-previous-link</code>	В заметке переход к предыдущей ссылке
C-c , n	M-x <code>action-lock-goto-next-link</code>	В заметке переход к следующей ссылке
C-c , x	M-x <code>howm-list-mark-ring</code>	Список меток-закладок
C-c , b	M-x <code>howm-list-buffers</code>	Вызов списка документов из буфера
C-c , Q	M-x <code>howm-kill-all</code>	Удалить все буферы (кроме несохраненных)

C-c , SPC	M-x howm-toggle-buffer	Переключение между документами howm/не-howm
C-c , t	M-x howm-list-todo	Вызов списка дел
C-c , T	M-x howm-insert-dtime	Вставка текущей даты и времени
C-c , y	M-x howm-list-schedule	Вызов расписания
C-c , d	M-x howm-insert-date	Вставка текущей даты
C-c , C	M-x howm-create-here	Создание датированного заголовка в длинной заметке
C-c , N	M-x howm-next-memo	Переход к следующему заголовку в длинной заметке
C-c , P	M-x howm-previous-memo	Переход к предыдущему заголовку в длинной заметке
C-c , L	M-x howm-last-memo	Переход к последнему заголовку в длинной заметке
C-c , H	M-x howm-first-memo	Переход к первому заголовку в длинной заметке
C-c , r	M-x howm-refresh	Активация ссылок на ключевые слова в новой заметке
C-c , R	M-x howm-menu-refresh	Обновление основного меню (необходимо, если у вас отключено автоматическое обновление)
C-c , M	M-x howm-open-named-file	Открытие файла (без режима howm)
C-c , w	M-x howm-random-walk	Автоматическое случайное пролистывание заметок

# 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.

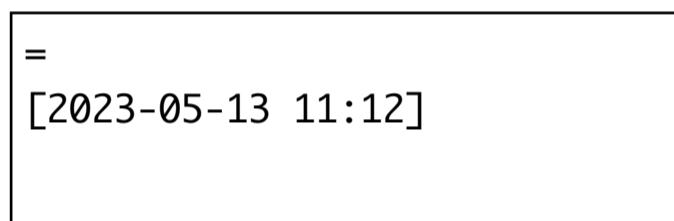
Ниже пошаговая инструкция овладения базовыми функциями howt.

## Шаг первый: создаем несколько заметок

Откройте Emacs

Используйте клавиатурный аккорд: C-c , c

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



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.