

Memo on the Electron

Ver2.0

tukasa

はじめに.....	3
知的生産の技術の歴史.....	3
その1 京大式カード.....	3
その2 Pile of Index Cards (PoIC)	7
その3 PCの検索機能を利用.....	9
その4 データベースからバッファへ.....	12
Memo on the Electron.....	14
基礎にあるアイデア.....	14
三つのフォルダ.....	14
Paper Plane xUI.....	16
実装.....	18
事前準備.....	18
MoE用設定.....	21
基本操作.....	25
運用.....	32
1. メモをためる.....	32
2. メモをカテゴリー分けする.....	33
3. メモをまとめる.....	33
ツール.....	36
記録メモ.....	39
変化の無いメモを別フォルダへ.....	39
論文メモ.....	40
応用編.....	42
参考文献・リンク.....	44

はじめに

Memo on the Electron (MoE) とは、私（つかさ）が開発したメモの方法論である。思いついたアイデアを手帳に書いた後、それをデータ化して、ファイラで管理するのが特徴である。基本的には、梅棹忠夫が京大式カードでやっていたことを PC でやっているだけだが、個々の箇所で大変な変更を施している。本冊子は、MoE の概要について解説したものである。

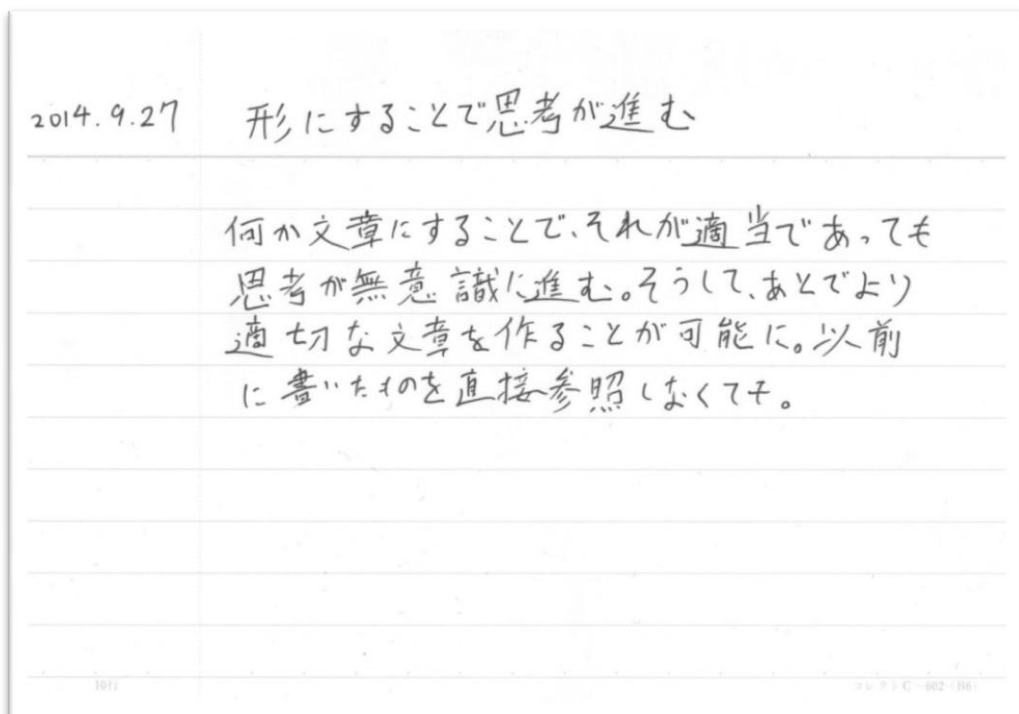
まずは、この手の技術に不慣れな人のため、知的生産の技術がこれまでたどってきた過程を歴史的に振り返ってみよう。

知的生産の技術の歴史

その1 京大式カード

1969 年初版の梅棹忠夫『知的生産の技術』が、この手の技術の走りである。そこには、京大式カードを使って知的生産を行う方法が記されている。

京大式カードの大きさは B6。紙は丈夫で、罫線が薄く引かれている。文房具店で、百枚入りのものが一つ 540 円程度で購入できる。記入の際は、日付、タイトルを必ずつける。一つのカードには一項目の内容のみを記述し、裏は使わない。そして、記入したカードはカードボックスに入れる。

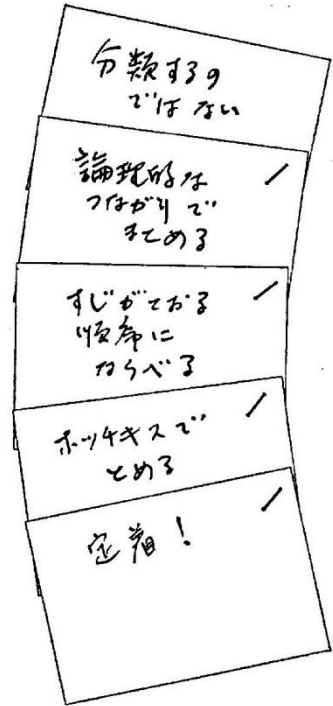


カードを使った知的生産

カードが相当数たまったら、それを「くる」ことによって発想を促す。見返してみたり、いくつか取り出して、見比べてみたりするのだ。そうすれば、思いもよらないアイデアが組み合わされることで、新しい発見をすることができるだろう。つまり、カードを「発想支援装置」として使うわけだ。

また、これは具体的に何か論文を書こうという場合にも使える。論文を作る時には、とりあえず頭にあるものをすべてカードに書き出す。そうしてから、カードを見比べ、関係するもの同士でまとめる。そうすれば、それは論文の個々の章と対応したものになるだろう。それを参考にしながら、論文を書いていくのだ。つまり、カードは「論文を作る時の素材」としても活用できるわけである(梅棹自身は、カードと別の紙切れを使ってやる方法を提案している。思いつきを紙切れに書き記し、論理的なつながりがあるもの同士を右図のようにホチキスでくっつける)。

操作できるというところが、カードの特徴なのである。蓄積と貯蔵だけなら、ノートで十分だ。ノートにかかれた知識は、しばしば死蔵の状態におちいりやすいので、カードにしようというのではなかったか。カードの操作のなかで、いちばん重要なことは、くみかえ操作である。知識と知識とを、いろいろにくみかえてみる。あるいはならべかえてみる。そうするとしばしば、一見なんの関係もないようにみえるカードとカードのあいだに、おもいもかけぬ関連が存在することに気がつくのである。そのときには、すぐにその発見をもカード化しよう。そのうちにまた、おなじ材料からでも、くみかえによって、さらにあたらしい発見がもたらされる。これは、知識の単なる集積作業ではない。それは一種の知的創造作業なのである。カードは、蓄積の装置というよりはむしろ、創造の装置なのだ。(梅棹忠夫『知的生産の技術』P57)



梅棹忠夫『知的生産の技術』P204

アイデアはバラバラのほうが使いやすい

京大式カードの基礎にあるのは、「アイデアは個々バラバラになっているほうが使いやすい」という思想である。そしてこの思想は、一般に用いられている文房具とは潮流を異にするものである。この手の技術に疎い人は、論文を書く時にはノートを利用するわけだ。ノートに思いついたことを書き連ねて、あとでそれを材料にして論文を作ろうとするだろう。だが、そのような試みはたいてい失敗する。ノートは、書き込んだアイデアを組み替えることができない。関係するもの同士を並列したいのなら、新しくノートを購入してそこに書き写すしかない。が、それは非常に手間がかかり、破綻してしまう。ノートには柔軟性がなく、知的生産に向いていないのだ。そこで、アイデアを個々バラバラに書けるようにし、最初から組み替えが可能ないようにしているのが、京大式カードを使ったシステムなのである。

ここにたどり着くには、思考作用は技術の問題であるという、醒めた視点が必要になる。自分が考えていることは有限であり、新しいアイデアも所詮は、過去に経験したことの組み合わせから生じるものでしかない。だから、アイデアをそれぞれ別個に書き出して、それを組み合わせれば、一つの思考ができあがる。精神作業とは、突き詰めれば、ただそれを効率化するものでしかない……という割り切った考え方が必要になるのだ。それは没個性的な技術であり、誰でも習得できるものであって、大思想家も深い思索を積んだ者も、他と区別されないことになるだろう。思考行為に神秘性を認め、そこに独自の価値を見出す

うとする人間にとっては、受け入れがたいことであり、普通では到達できない境地なのだ。たいていの人は、自分の知的生産がうまくいかない場合、その責任を自身に帰す。論文が書けないのは自分の頭が悪く、精神的に未熟だからと反省し、さらに思索を深めるなり、偉大な先達の原典を読み込むなり、といった努力をすることになるだろう。だが、本当はそうでは無いのだ。論文を書けないのは、単に自分のやり方が悪いからである。問題点は技術的な未熟さにある、と総括し、それを突破しようとしているのである。カードを知的生産に使うことは、それゆえそれまでの方法論とは一線を画しており、飛躍なのである。

技術というものは、原則として没个性的である。だれでもが、順序をふんで練習してゆけば、かならず一定の水準に到達できる、という性質をもっている。それに対して、研究だとか勉強とかの精神活動は、しばしばもっとも个性的・個人的ないとなみであって、普遍性がなく、公開不可能なものである、というかんがえかたがあるのである。それは、個性的な個人の精神の、奥ぶかい秘密の聖域でいとまれる作業であって、他人にみせるべきものではない……。

しかし、いろいろとしらべてみると、みんなひじょうに個性的とおもっているけれど、精神の奥の院でおこなわれている儀式は、あんがいおなじようなものがおおいのである。おなじようなくふうをして、おなじような失敗をしている。それなら、おもいきって、そういう話題を公開の場にひっぱりだして、おたがいに情報を交換するようにすれば、進歩もいちじるしいであろう。そういうようにしようではないか、というのが、このような本をかくことの目的なのである。(梅棹忠夫『知的生産の技術』P8)

カードの死蔵

このシステムは、取り扱うカードが少ないうちは問題なく機能する。レジュメを作りたい時は、まずは思いついたことなり調べたことなりを、片っ端からカードに書いていく。書き終えたらカードを並び替え、それを材料にして文章を書く。白紙を前に考え込みながら、いきなり漠然と書き始めるよりは、格段に質のいいものが短時間でできるはずだ。

だが、複数の事柄について書かれたカードが、カードボックスの中に大量に蓄積されるようになると、このシステムは機能しなくなる。例えば、ある特定のテーマの論文を書く必要がある時、以前に書いたメモを参考にしようと思って、カードを見返してみるとしよう。すると出てくるのは、そのテーマとは無関係な、個々バラバラの内容について書かれているカードだ。読んだ本のメモ、経験したことの記録、ずっと以前に取り組んでいた課題に関する思いつき……。出会い、経験し、取り組んでいる課題の多様さにあわせて、カードボックスには様々な種類のカードが蓄積されている。そして、カードの多様性に比例して、現在取り組んでいるテーマに関係するカードを取り出すことが、難しくなるのである。さらにここに、時間経過によるメモの劣化という問題が加わる。アイデアメモは、それを思いついた当初に持っていた問題意識が失われると、意味が取れないものになってしまうのだ。カードをめくり、古い方へ行けば行くほど、理解しがたいカードに出会うことになるだろう。結果、知的生産をするときには、最近作成したカードだけをめぐってみて、今の問題意識に関係がありそうなものを取り出すことになるだろう。そして、過去のカードは、今の問題意識からかけ離れた理解できないものとして、全く顧みられなくなるだろう。これは、それらを一つのデータベースとしてためていることが、無意味だということを意味する。過去に書いたカードが、死蔵してしまうのだ。

発想支援装置にもならない

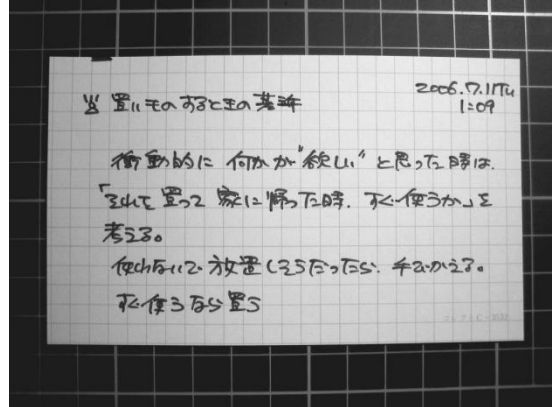
さらに、アイデア発生装置としても、このデータベースは役に立たない。たとえば、私が過去に使っていたカードボックスには、当時ハマっていたFPSの特定マップの攻略アイデア、気になったレシピ、何かの思いつきの断片、何かの記録、といったものが混じっている。さらに、過去のデータになればなるだけ、意味の取れなくなったアイデアの欠片に出会うことになる。これらを見比べて、何か意義のあることを思いつけるわけがない。

複数の情報を並列して発想するには、それぞれが洗練され、かつ一定の関連性を持っている必要がある。アイデア段階で打ち捨てられたものが無秩序に並んでいても、意味がないのである。新しい刺激が欲しいのであれば、誰かに話を聞きにいきなり、町に出て新しい経験をするなり、誰かの本を読むなりすればいい。多大な手間をかけて、自分でデータベースなどを作る必要は無いのである。

その2 Pile of Index Cards (PoIC)

梅棹の京大式カードの次の段階に来るのが、PoICである。

PoICでは、5×3の方眼のカードを利用する。タイトルの横についているのは、カードのジャンルを分かりやすくするためのアイコンで、これは「発見」アイコンである（他に「記録」「GTD」「参照」アイコンがある）。左上隅の塗りつぶしの位置は、アイコンと対応しており、カードボックスから目的のカードを見つけやすくしている。他にも、カードを持ち運ぶ方法について考察していたり、アイデアを書き留める野帳の使い方についても書いていたり、考察者の実際の試行錯誤に基づく修正が盛り込まれている。梅棹の著書は具体的な方法論について詳しく論じたものではないので、カードを試してみたい人はこれを参考にするのがいいだろう。



タスクフォース

その中で最も重要な過程の一つに、タスクフォースがある。それは、ためたカードを元に再生産をするプロセスである。広い場所を用意し、そこにカードを並べていく。そして、カード同士を見比べ、似た内容のカード同士を束ねてグループ分けし、グループの内容を記した付箋を貼る。このカードの束を元にして文書を作る。

文書に内容を反映した束は、お役御免として、カードボックスとは別の場所にしまう。そして、カードボックスには、使わなかったカードだけを戻す。こうして、カードボックスからは利用したカードが消え、カードボックスの中身は、タスクフォース前よりも秩序だったものになる。

タスクフォースの画期性

ポイントは以下の二点である。

1. 再生産の過程を組み込んだこと
2. カードボックス内の秩序を一定に保つ方法論を提示したこと

これは、梅棹忠夫の京大式カードと似たような方法に見えるかもしれないが、実は思想的に逆を行くものである。梅棹の場合、築くのはデータベースである。カードがたまればたまるだけ、システムの有用度は高まるというのが根本思想なのだ。それゆえ、「利用したカードを取り除く」といった、データベースを破壊する操作は原理的にできない。たとえカードを活用したとしても、それは必ずあとで元の位置に戻すのである。梅棹の思想では、カードによる知的生産がうまく機能しないとしたら、それはカードの枚数が足りないからだ、というように総括されるわけだ。

PoICは、京大式カードの限界点を、「アイデアから再生産する過程」を意識的に作り出さなかったことに求めている。カードの死蔵が起こるのは、時間が



タスクフォース。左が通り抜かれたタスクフォース。右が残りのカード(拡大する) 63



グループ化(拡大する) 63



着付け(拡大する) 63



空欄配置(拡大する) 63



コンパイル(拡大する) 63

経過しカードが増えることによって、カードボックスがカオスになるからだ。そこで、カードを比較し、束ね、廃棄する、というタスクフォースの過程を組み込むことで、カードボックス内の秩序を一定にしようとしている。この点で梅棹の方法論とは別物であり、思想的な飛躍が存在するわけである。

時系列でカードを蓄積していくと、自然の法則に従って、システムの中のエントロピー（情報の乱雑さ）は一方向的に増えていきます。分類しない時系列では、なおさらです。このままでは、PoIC は破綻しそうにも思えます。私自身、カードが増えるにしたがって、このまま行ったらどうなるのだろうか、と心配になったことがありました。

この自然の法則に逆らってエントロピーを減らそうとする場合、人間の「努力」が必要になります。図書館や博物館では、「つねに分類する努力」によってこれを実現しています。そのために、これらの公共施設では高いコスト（人件費、時間）を払っています。しかし、前述のように、PoIC では積極的に（？）検索・分類しません。では、どのようにしてシステムの破綻を防ぐのでしょうか。

答えは簡単で、やはり検索・分類するのです。従来の方法と違うのは、これが一番最後に来ることです。PoIC において、カードを書くのは、個人の知識のデータベースを構築することです。しかし、これはまだ準備段階です。PoIC の本当の目標は、このシステムを使って、新しい知恵・知識・成果を再生産することです。そうして初めて "Get things Done!" となります。(PoIC - 時系列スタック法)

タスクフォースの限界

私は梅棹の京大式カードで挫折したあと、PoIC に影響されて、タスクフォースの過程を取り入れた。だが、それでもやはり、死蔵という問題は解消されなかった。カードという物理的特性がネックになるからだ。

タスクフォースを実行するには、まず、広い場所を用意しなければならない。気楽に机の上でやることはできない。少なくとも畳数枚程度の広さが必要になる。また、この作業は一度に行わなければならない。そうそう何度もカードをすべて調べて並び替えることはできないし、時間が経過すると、どのカード同士が同じ内容なのかの判別もできなくなってしまう。カードがたまるにつれて、負担が増大することになるのだ。

再生産が必要だ、という PoIC の発想は正しいかもしれない。だがそれは、カードを用いるという前提自体が壁になることで、行き詰まるのである。ここに、それを乗り越えるものとして、PC を用いる方法が要請されることになる。

※画像はどちらも PoIC のホームページから引用

その3 PCの検索機能を利用

PCを用いて知的生産を行おうという試み自体は、以前からある。アウトラインプロセッサのような、それ専用のソフトは数多く存在するし、howmやChangeLogのように、エディタの拡張として実装したものもある。私は、その中で、テキスト形式でメモをためる方法を追求した。メモファイルが、特定のソフトでしか開けないものであった場合、後々そのメモが使えなくなることを恐れたからだ。

そこで期待したのは、PCの機能の有効活用である。例えば何か調べ物をしたい場合、Googleで適当な語を検索すれば、自分に必要な情報が乗っかっていそうなリンクがずらっと表示され、それをいくつか開けたい場合、疑問は解決する。はてなやwikiを読んでいてよくわからない言葉があれば、その語のリンクをたどれば、その説明を見ることができる。ニコニコ動画で興味のある動画に出会ったら、それに付けられているタグをクリックすれば、関連する動画が羅列して表示される。このように、ネットにおいては、データが無数にあったとしても、そこから自分にとって有用な情報を、PC特有の機能で取得する方法が存在する。これを応用すれば、カードで試行錯誤していた「情報が増えることによるカオス化」など、容易に解決するのではないか、と思ったのだ。

移植

だがその前に、京大式カードと同じことをPC上でもできるようにする必要がある。エクスプローラーとメモ帳にしか触れたことが無く、「メモの中身を確認するために一々ファイルを開くのか面倒じゃないの?」と思っている人がいるかもしれないので、ここで説明しておこう。

京大式カードでは、カードを「くる」という操作を行っていた。この操作は、カードのタイトルだけをざっと見ることで関係するものを見つけ出す操作と、カードの本文をざっと見ることで関係するものを見つけ出す操作の二つからなっている。これをPC上で実現するために必要になるのは

- タイトル一覧表示
- 連動ビュー

の二つである。前者が、「カードのタイトルを見る」に対応する。あるフォルダにメモファイルが入っていると、各メモのタイトルが一覧表示される仕組みがあればいい。後者が、「カードの本文をざっと見る」に対応する。ファイルにカーソルをあわせれば、それと連動して、別ウィンドウでそのファイルの中身が表示されればいいわけだ。これを実現する方法は、TextTree、howm等いくつかある。これで、カードで実現していた「くる」操作は可能になる。

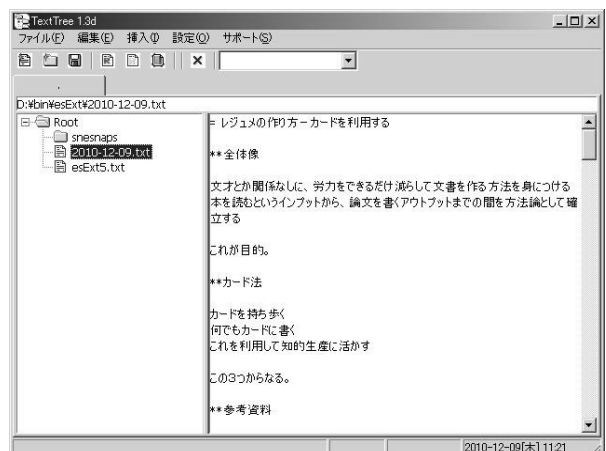
Grepによる関連メモの並列

そこで本題の「PCの機能を利用して、データベースから関連するメモを抜き出す」に移る。これの追求で、私は立ち止まった。

それには

- Grepの利用
- 関係するファイルに共通のタグを入れる
- データ本体とは別に情報管理

の三つの方法がある。



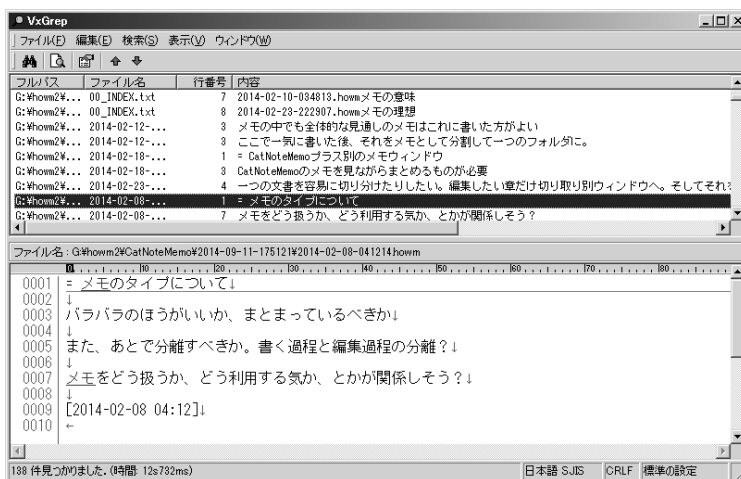
TextTreeによるメモの管理

まずは Grep について話そう。Grep は、複数のテキストファイルを対象にして、特定の単語を検索するプログラムの名称である。これは、エディタの機能としてついていることが多い。また、Grep のみに特化したソフトも存在する。だが、Grep を利用して自分が知りたいテーマについての情報を並列するのは難しい。そもそもテーマに関係する単語が存在するののかという問題がある。また、たとえあったとしても、それによってメモの内から関連するものをすべて拾える保証がない。それに、この方法だと、どうしても関係ないメモがヒットしてしまう。

この方法は、Google で調べものをした際の類推から、有効なものであると思われる。例えば何か調べものがあるとき、我々は Google で、それに関係していそうな単語を検索する。すると複数のサイトがヒットするので、それを上から順に開いていく。そうすれば、たいいはいは知りたいと思った情報に行き着くわけだ。

だが、Grep をかける対象が未完成なアイデアである点が、Google 検索の状況と大きく異なっている。Google 検索をかける対象は、完成したデータである。だから、ヒットしたリンクには、知りたい情報が載っている可能性が高い。しかし、その対象がアイデアメモの場合、ヒットするのは未完成な情報である。それ単体では無意味なものでしかなく、それをいくつ閲覧したところで知りたいことは書かれていない。さらにそれは、過去のものになればなるほど、それを書いた際の問題意識の忘却から、内容が理解できないものになる。それだけならまだしも、全く現在の問題と関係の無いメモまで同時にヒットしてしまうのだ。だから、Grep によって自分の求める情報が手に入る可能性は非常に低い。

VxGrep で「メモ」という語を検索



関係するファイルに共通のタグを入れる

次にタグ付けについて。この場合、ファイル名検索でヒットするよう、次のようにタグをつけることになる。

【タグ】ファイル名.txt

重要度を記号で表す場合も、原理的にはこれと同じである。例えば、ファイル名の末尾にスターをつける。

ファイル名☆☆☆.txt

このようにして、タグなり記号なり、共通の単語をファイル名に入れる。そうすれば、その単語でファイル名検索をした時、関連するファイルを一覧表示できるわけだ。

このときの問題点の一つは、持続性を持つタグをつけることができない、ということである。アイデアメモを作った当初は、それが何についてのアイデアかをカテゴリズできていない場合が大半である。それゆえ、現時点の判断で、将来までずっと通用するようなタグを付けることは、ほとんど不可能だろう。だからどうしても、おおざっぱなタグか、あるいはそのときにたまたま思いついたタグを付けることになる。

「おおざっぱなタグ」は、おおざっぱなゆえに役に立たない。そのときどきの問題関心は移り変わる故に、同じタグであっても、メモ同士の関連性は薄いものになる。メモを利用するには、自身の問題意識に則した細かな区分が必要になるのだ。

「そのときにたまたま思いついたタグ」は、持続性が無いゆえに役に立たない。ある日思いついたアイデア A に、思いつきで「ほげほげ」とタグを付けたとしよう。しかし、次にそのアイデア A と関連するアイデア B を思いついた時、「前に似たようなアイデア思いついたよな」ということは覚えていたとしても、それに付けたタグを正確に思い出すことができない。そうして、そのときたまたま思いついた、また別のタグを付けることになってしまう。しかし、タグで検索するからには、タグ名は正確に一致していなければならぬ。だから、このようにして付けたタグは無意味になる。

また、タグを付けても、重要度での区別をつけることができないという問題がある。メモの中には、他の複数のメモを参考にして書いた、完成度の高いメモと、ただ思いつきを書いただけの、完成度の低いメモが混在している。しかしそれらは、タグで検索した時、すべて同じように表示されてしまう。有用なメモが、他の雑多で低価値のメモの中に埋もれてしまうのである。同じタグを持つファイルが増えれば増えるほど、この事態は顕著になる。結局、カードボックスの中で、有用なカードが大量の他のカードに埋もれて死んだのと同じように、同じタグを持つメモファイルの中で、有用なメモファイルが死んでしまうことになる。

データ本体とは別に情報管理

次に「メモデータ本体とは別に情報管理」について。カード型データベースソフトを使い、一つのデータに対し一つの蔵書カードを作成。検索その他はこの蔵書カードを利用して行う。気になって調べはしたが、適当なカード型データベースソフトを見つけられなかったという理由であきらめる。

そして破綻へ

結局、メモをデータベース化して PC 特有の機能で操作する、という試みは破綻した。結果、最近のメモのみを一覧表示し、関係するメモをいくつか見比べる、という操作のみをすることになった。これは、最近作成したメモ以外はあってもなくてもいいということ、すなわち過去のメモ作成にかけた労力が無駄になったということの意味する。死蔵というカードと同じ問題が、ここでも生じたわけだ。

その4 データベースからバッファへ

野口悠紀雄のバッファ理論

PC に特有の機能を使っても、問題は解決しなかった。いくらメモをためたとしても、それを有効に使う方法はない。データはやはり死蔵してしまったのだ。既存のツールを使うだけではだめだ。知的生産を実現するためには、この状況を突破する必要がある。

その突破のきっかけになったのが、野口のバッファ理論だった。これは、野口悠紀雄『「超」整理法〈3〉』において出てくる理論である。

知的生産活動で扱うデータは、決まりきった仕方では処理できない、という特性を持っている。それは、最初は何のカテゴリーに入るかも不明で、重要度も分からない。それを扱う方法に先例もモデルも無い場合が大半だ。このような性質を持つデータに対しては、処理が確立されている、定型的な仕事でとられてきた方法は使えない。今までとは異なる発想に基づく処理システムを用意し、そこで処理をすべきである。野口はこのような考え方をする。

まず、対象がフロー（流れ）であることを明確に意識する必要がある。必要とされるのは、膨大な量のフローを制御するダイナミック（動的）な方法である。一定量のストック（蓄積）を管理するスタティック（静的）な対処法ではない。

これは自明のことである。しかし、従来の収納システムは、「内容がほぼ変わらないストックのための管理」のためのものだ。「大量のものが流入し、ストックの内容が短期間のうちに入れ替わってしまう」という認識は、殆どないのである。（野口悠紀雄『「超」整理法〈3〉』P21）

ではどうするか。それには、「とりあえずおいておく場所（バッファ）」を用意し、そこに分類せずにデータを入れればよい。ある程度の時間がたてば、そこに放り込まれたデータは自然に醸成し、区分も判明になるだろう。そのときになってはじめてそのデータを取り出し、分類して、処理をすればいい。

マゼラン的な仕事を扱うには、マニュアル遵守的な仕事とは異なる発想にもとづいて、処理システムを構築する必要がある。

とくに重要なのは、「バッファ」(buffer)だ。これは、「緩衝器」、つまり二つのもの、あるいは二つのプロセスの中間にあって、衝撃を受け止めるための装置である。

まず、外から入ってきたものや新しく作ったものを、簡単な手続きによって（できれば、殆ど手間をかけずに）、システムの中に受け入れる必要がある。整然とした収納でなくともよい。しかし、書類が紛失したり迷子になったりすることはないようにする。つまり、「とりあえず受け入れる」のである。これが、「受け入れバッファ」だ。

そして、不要と思われるものを必要なものから区分し、所要の措置や加工などを行い、次の段階に送る。

この際、確立された処理法はないのだから、本当に正しい処理をしたかどうかは分からない。やり直す必要があるかもしれない。そのため、完全でなくともよいから、一応の措置をする。一〇〇%処理を目論んで何もしないのではなく、とにかく一歩進める。「ゼロか完璧か」でなく、八割の処理をするのだ。これは、単なる先送りとは違う。「もっとも重要と思われること」は、行っておくのである。

そして、「多分必要ないだろう」とと思われるものを、日常の仕事のジャマにならないようなところに置く。つまり「とりあえず捨てておく」のである。これが、「廃棄バッファ」だ。（野口悠紀雄『「超」整理法〈3〉』p41）

バッファによる解決

アイデアメモが使いえなくなったのは、メモが処理しきれなくなるほどたまり、古いデータがノイズになって、全体が機能しなくなったからである。そこで、発想を転換する。アイデアメモをためる場所は、データベースではなく、バッファだ。そこに入れたデータは、一時的に保存されているだけである。時間が経過して、それらのアイデアメモがどのような問題意識によって書かれたものかが判然としたなら、それらをまとめて別の場所に移動し、その上で知的生産作業を行う。分類は、後でするのだ。

- 最初は区分せず、とりあえず入れる場所を用意してなんでもそこに放り込んでいく
- 半自動的にそれが保てるシステムを構築する

この二点を実現すれば、不定型なデータの処理が可能になる。

今までアイデアメモをうまく活用できなかったのは、データベースの規模の問題でもなければ、データベースの運用方法に通じていないからでもない。前提としていた、データベースを作る、という発想からして既に間違っていたのだ。個々のものの重要性がはじめからわかっており、カテゴライズがすでになされているようなものであれば、データベースを築くことは有効である。それは、規模を増せばそれだけ、有用度を増すだろう。だが、アイデアのような不完全なデータを大量にためても、無意味なのである。それをもとにしてデータベースを築いたとしても、アイデアは個々バラバラのスピードで陳腐化していく。結果、データベースはカオス化し使いものにならなくなる。この問題は、データベースから有用なデータを抜き取る機能をいくら追求したとしても、決して解消されないだろう。それよりは、バッファが半自動的に洗練され、不要なものが削ぎ落とされ、常に自身にとって有益な情報がある状態を保ち続けることのほうが、重要なのである。

このことに気づいていなかったから、PC 上でアイデアメモを管理する試みは失敗していたのである。そこでは、どうしてもデータベースを否定するような発想、例えば「過去のデータを編集する」「不要なデータを消す」「不可逆的な仕方特定のグループに分類する」といったことができなかった。そして、アイデアメモを有効活用するための試みも、タグ付けといった表層的な操作に限定されていた。我々は、既存の思考に縛られすぎていたのだ。ノートを捨ててカードを使い出した時のように、データベースを捨て、バッファを中心とした方法を構築する必要がある。

Memo on the Electron

データベースを作る必要はない。だから、メモが不要になればそれを捨ててしまってもかまわない。自由にデータを組み替え、カテゴライズしたものを別の場所に移動し、原本を改変せよ。全体が有用であり続ける状態をつくれ……。この発想を元にして、MoE の原型はできあがった。あとは、これを PC で実現すればいいわけだ。

基礎にあるアイデア

- 最初は区分せず、とりあえず入れる場所を用意してなんでもそこに放り込んでいく
- 半自動的にそれが保てるシステムを構築する

の二点を PC 上で実現するために、以下のことを行う。

1. 特定のフォルダを作成し、それをバッファとして利用する。そこにテキストファイル形式で、メモをためていく
2. 時間が経過したら、そのフォルダにあるメモを、問題関心で区分して別フォルダに移動する
3. 不要なメモが見つかったら、それを廃棄用のフォルダに移動する

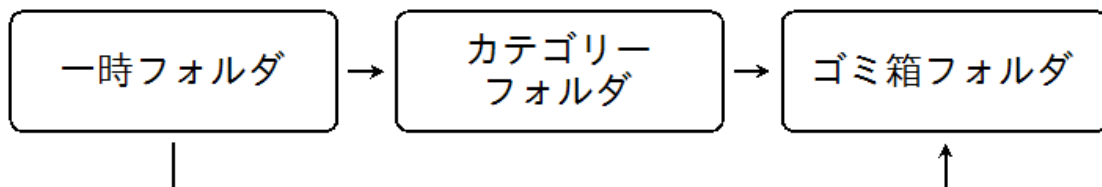
フォルダから関連するメモを取り出し、別フォルダに移動するプロセスは、関連するメモ同士のまとまりを作るという意味だけではなく、バッファの秩序を保つという意味も持つだろう。また、メモを廃棄するためのフォルダをあらかじめ用意しておく。不要だと判断したメモを容易に捨てられる状況を作るとは、全体の秩序を維持することにつながるだろう。

三つのフォルダ

では次に、フォルダ構造とファイル移動の観点から見てみよう。必要なのは、

- 一時フォルダ……最初にメモファイルを保存するためのフォルダ
- カテゴリーフォルダ……カテゴリー分けのためのフォルダ
- ゴミ箱フォルダ……不要なメモを捨てるためのフォルダ

の3つのフォルダである。メモファイルを、以下の図のように順番に移動することで、常に有用なメモのみを利用できるようにする。

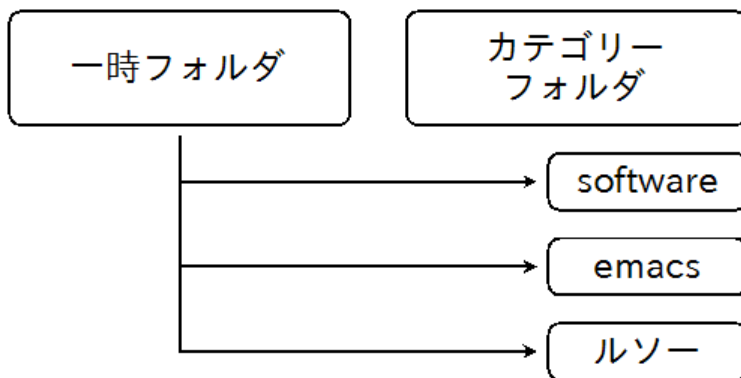


1. 一時フォルダにためる

アイデアメモは全て、一時フォルダに無差別に保存する。メモの種類も、カテゴリーも、以前に書いたメモとの関係性も一切気にせず、どんどん思いつくままに作っていきこう。

2. 一時フォルダからカテゴリーフォルダに移動

時間が経過すれば、各々のメモが、どのような問題意識で書かれたものか明確になってくる。そうしたら、そのとき初めてカテゴリー分けを行う。カテゴリーフォルダに新たにフォルダを作成し、関係するファイルをまとめて移動する。個々のメモの内容は、タイトル表示とビューアで確認すればいい。



こうして、カテゴリーフォルダの中に、個々のカテゴリーフォルダを作っていく。後は、各々のフォルダへ行ってメモを閲覧しながら、知的生産をすればいいわけだ。

3. ゴミ箱フォルダへの移動

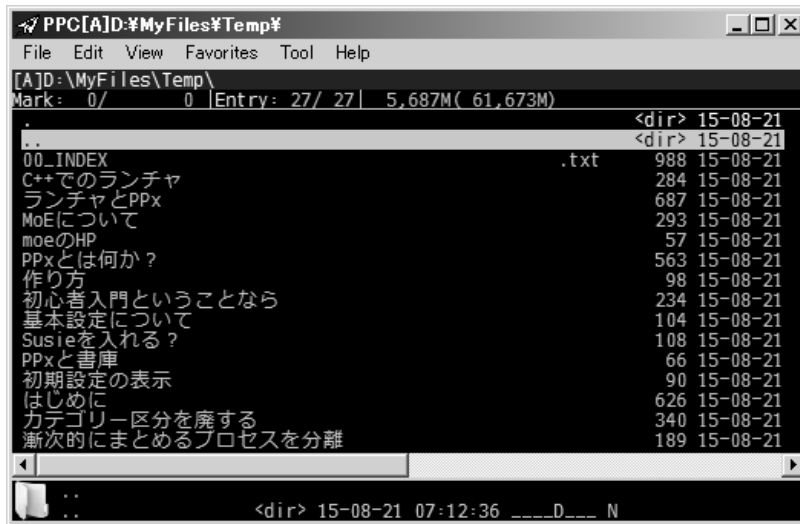
一時フォルダ内やカテゴリーフォルダ内のメモの内容を、ビューアで閲覧している時に、不要になったメモが見つかることがあるだろう。例えば、作成した文書に内容が反映されているメモや、状況の変化によって、すでに問題意識自体が古くなっているメモだ。それを見つけたら、その都度、ゴミ箱フォルダに移動する。フォルダ単位で不要だと判断したものがあれば、それも同様に、ゴミ箱フォルダに移動する。

※ここでは便宜上、一時フォルダとカテゴリーフォルダを分けて説明しているが、実際の運用では、一時フォルダの下に個々のカテゴリーフォルダを作ることになる。そうすると、一時フォルダに行けば今取り組んでいる課題がひと目でわかることになり、効率がいいのだ。

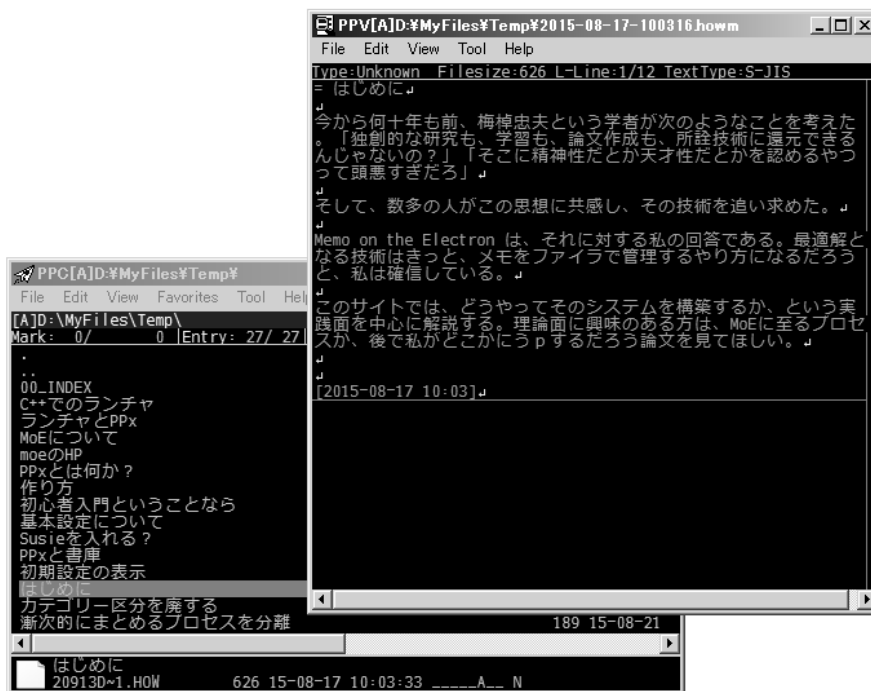
Paper Plane xUI

ファイル操作には Paper Plane xUI(PPx)を使う。

PPx からエディタを起動し、一時フォルダにメモファイルをためていく。



メモファイルがたまったら、ビューアで内容を確認しながら、関係するメモファイルをカテゴリーフォルダに移動する。また、不要なメモファイルを見つけたら、それを適宜ゴミ箱フォルダに移動する。



ひらめきを集積すること

人間は、常に複数の課題に取り組んでいる。それは、友人や仕事に関するような恒常的なものであったり、部屋の片付けをどうするか、最近携帯の調子が悪い、食材を帰りにどのスーパーで買うかというような、一時的なものであったりする。そのうちの一つに意識を集中すれば、それへの対処法、方針、総括、感想といったものが、無秩序に、断片的にひらめくことになる。だが、すぐにまた別の課題に思いが移ることで、ひらめきは断片のまま忘却されてしまう。そうして、何度も同じひらめきを繰り返し、忘却してというように、同じところをくるくると回ることになるのである。

そこで、思いついたことをメモにする。ひらめきの断片を集積することで、課題は次第に輪郭をはっきりさせ、それにどう対処すればいいか、そのための手続きは具体的にどうなるのかが明確になるわけだ。そうすれば、それまで断片的な思いつきのみで課題に取り組んでいた時よりも、適切な行為ができるようになるだろう。

我々は現実において様々な課題に出会い、そのときどきでなんらかの対処をする。その実際の流れとは別に、アイデアを一旦、メモ群という別の場所に保存し、寝かせたあとに、それを再び現実に反映させるのだ。

バッファ理論とファイラ

バッファ理論に基づいて、死蔵しないシステムを構築するには、メモファイルの柔軟な移動が必要になる。それを実現するには、任意の場所で、ファイルあるいはフォルダを、作成／削除／移動／リネームできなければならない。メモのデータベースを築く際は不要であった、ファイラと同等の機能が要求されるわけだ。こうして、PC を用いたメモシステムは、さらに条件を限定されることになる。それは、「連動ビューとタイトル表示を可能にするもの」であり、かつ「ファイラをベースにしたもの」でなければならない。そして、この条件のもとに私が築いたシステムが、MoE である。

ここに至るには、「知的生産の技術」と「ファイラ」という、あまり関連性のない 2 つの知識を、共にある程度のレベルで兼ね備えていなければならない。これが、「PC を知的生産に利用する」という、誰でもすぐに思いつきそうなアイデアが、なかなか実現されてこなかった理由ではないかと私は思う。私は、たまたまファイラのカスタマイズを趣味としていて、かつ Paper Plane xUI という、おそらくはタイトル表示を唯一できるファイラを愛用していたから、MoE を実現できたのである。

実装

では、実装する方法について具体的に説明していこう。ちなみに、MoE のホームページ (<https://sites.google.com/site/moesystem/>) にもほぼ同じ内容のことが書かれている。コードのコピーやスクリプトのダウンロードは、そちらを参考にして行ってほしい。

事前準備

まずは、ファイラとエディタの設定をする。

ファイラ (Paper Plane xUI)

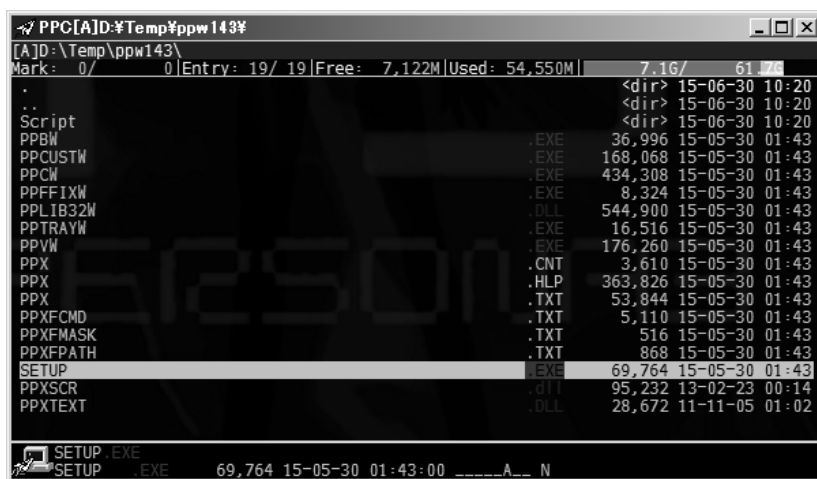
TORO's Library (<http://toro.d.dooo.jp/slppx.html>) に行く。そこで

- PPx 本体
- PPx Script Module
- PPx Text Module

をダウンロード。続いて

1. PPx 本体を解凍
2. PPx Script Module を解凍し、中にある PPXSCR.dll (64bit 版なら PPXSCR64.DLL) を PPx フォルダにコピー
3. PPx Text Module を解凍し、中にある PPXTEXT.dll (64bit 版なら PPXTEXT64.DLL) を PPx フォルダにコピー
4. PPx フォルダ内に、Script という名前で新しいフォルダを作成

以上で、PPx フォルダは次のようになる。



次に PPx フォルダにある SETUP.EXE を実行。セットアップ方法は「おまかせインストーラ」を、インストール先の選択は「コピーなしでインストール」を選択。後は適当で大丈夫だ。

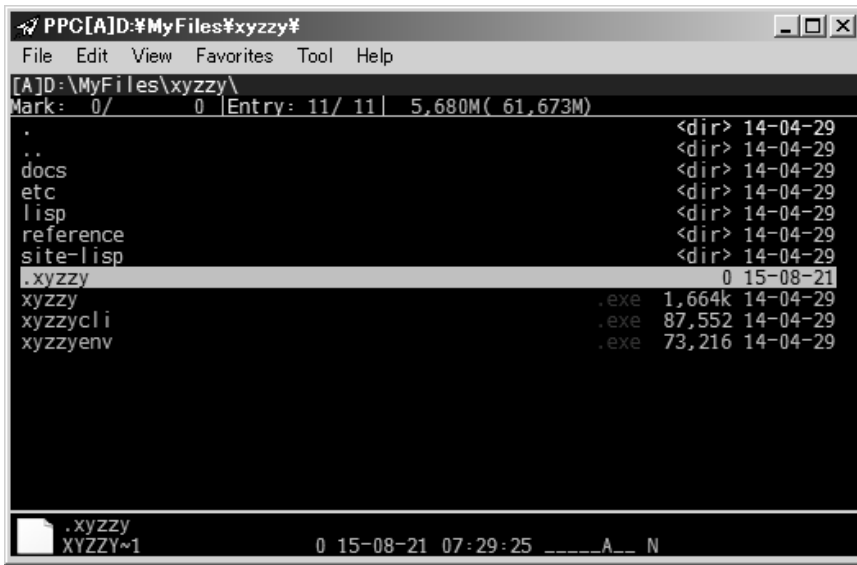
エディタ (xyzyz)

github 版 xyzyz 配布サイト (<http://xyzyz-022.github.io/>) に行く。そこで xyzyz をダウンロードし、解凍する。

次に、xyzyz フォルダ内の site-lisp フォルダに siteinit.l を作成する。何も無いところを右クリックし、「新規作成 - テキストファイル」を選択。siteinit.l と入力し、OK を押せばいい。



同じ要領で、xyzyz フォルダに「.xyzyz」を作成する。



ついで、環境変数の設定を行う。コントロールパネル→システムとセキュリティ→システム→システムの詳細設定→環境変数→ユーザー環境変数の新規をクリック。変数名に XYZZYHOME を。変数値に xyzyz フォルダのパスを入力して OK を押す。



MoE 用設定

ここからは、MoE を行うために必要な独自の設定である。以下の5つを順番にやっていけばいい。

① howm-create2dir.l (xyzyy)

howm ファイルを作成するスクリプト

howm-create2dir.l

を xyzyy の site-lisp フォルダに入れる。

② xyzyy の設定ファイル

xyzyy フォルダの .xyzyy をエディタで編集し、以下を追加する。PPx では、エントリにカーソルをあわせて E を押せばデフォルトのエディタで、Shift+E を押せば PPe で編集することができる。

```

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;;howm モード

(pushnew '("\.howm$" . howm-create2dir-mode) *auto-mode-alist* :test #'equal)
(require "howm-create2dir")

```

③ MakeComment.js (PPx)

PPx の Script フォルダに、次のファイルを保存する。

MakeComment.js

MakeComment.js は、タイトルとファイル名とを対応させたコメントファイル(00_INDEX.txt)を、フォルダ内に自動で作成するスクリプトである。

④ カスタマイザーで編集して取込 (PPx)

以下のコードを、クリップボードにコピーする。

```

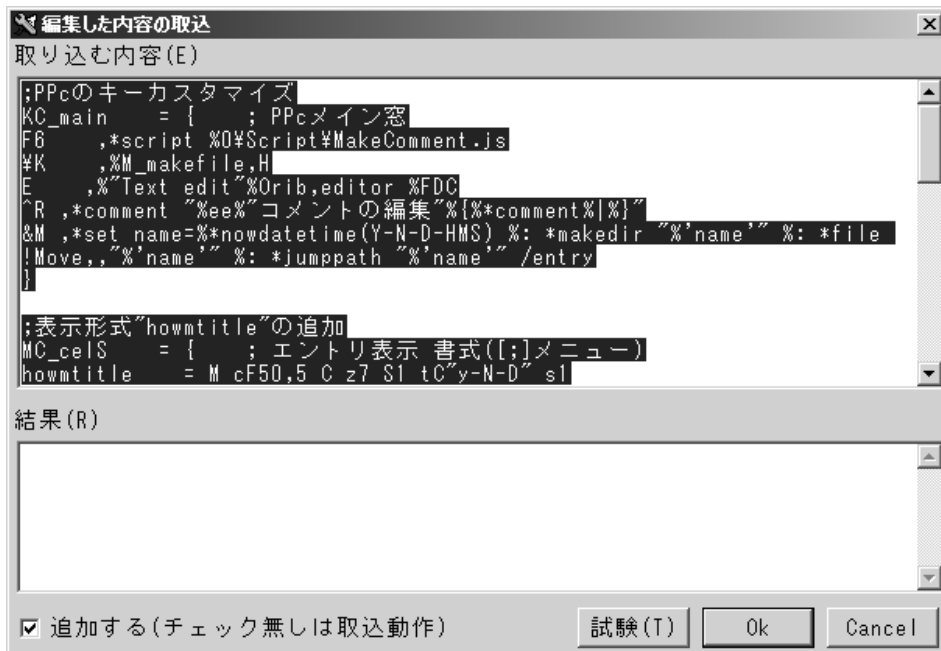
;PPcのキーカスタマイズ
KC_main = { ; PPc メイン窓
F6 ,*script %0\Script\MakeComment.js
\K ,%M_makefile,H
E ,%"Text edit"%Orib,editor %FDC
^R,*comment "%ee%"コメントの編集"%[*comment%|%"
&M ,*set name=%*nowdatetime(Y-N-D-HMS) %: *mkdir "'name'" %:
*file !Move,,"'name'" %: *jump path "'name'" /entry
}
;表示形式"howmtitle"の追加
MC_celS = { ; エントリ表示 書式([:]メニュー)
howmtitle = M cF50,5 C z7 S1 tC"y-N-D" s1
}
;拡張子が howm の時に Enter を押すとビューアが起動
E_cr = {
HOWM ,%v "%1%\%R"
}
;PPvのキーカスタマイズ
KV_main = {
UP ,%KC"@UP@N"
DOWN ,%KC"@DOWN@N"
LEFT,*execute C,*cursor 6, -1 %: %KC"@N"
RIGHT,*execute C,*cursor 6, 1 %: %KC"@N"
SPACE ,%KC"@SPACE@N"
\SPACE ,%KC"@\SPACE@N"
ENTER = @Q
}
;howm 作成用メニュー
-|M_makefile =
M_makefile = { ** comment **
&howm-create2dir = %Ob,xyzyy -e (howm-create2dir) %1
-- =
ディレクトリ(&D) = %K"@K"
}
XC_cwrt = 2 ; コメントファイル作成は 0:手動 1:確認有 2:自動
;フォルダ名を日時から付ける
Mes0411 = {
NWDN = %*nowdatetime(Y-N-D-HMS)
}

```

PPx の上部にあるメニューバーから「ツール — カスタマイズ」を選択するか、PPx フォルダの PPCUSTW.EXE を実行するかして、カスタマイザーを起動。「編集して取込」をクリックする。



すると、「編集した内容の取込」ウィンドウが、クリップボードの内容をペーストした状態で表示されるので、OK ボタンを押す。これで、このコードを取り込むことができる。

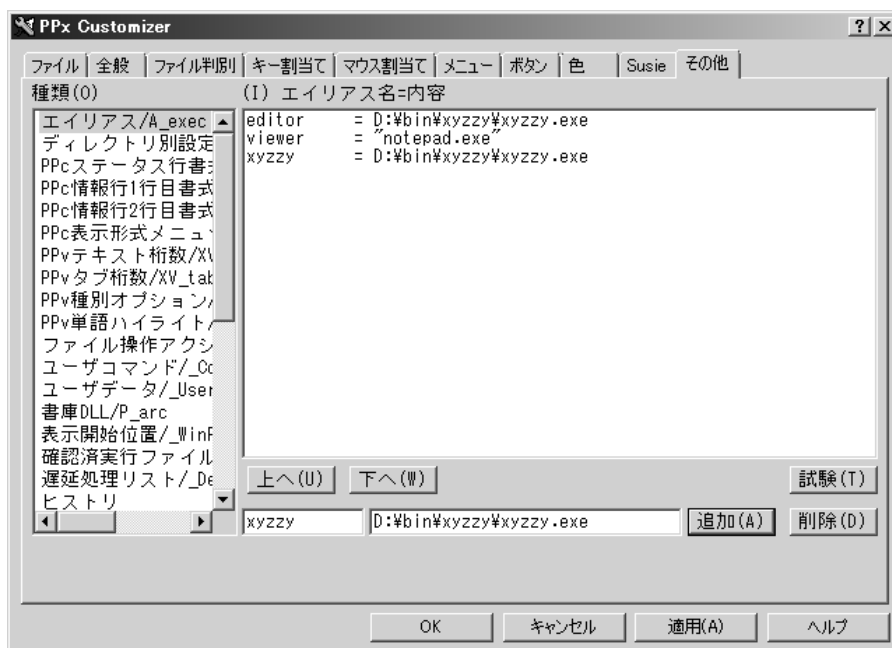


⑤ エディタの設定 (PPx)

カスタマイザーを開き、その他タブのエイリアスから、editor のパスを xyzzy のパスに変更し、追加ボタンを押す。



次に、エイリアス名に xyzzy、内容に xyzzy のパスを入力して追加を押す。上の欄に追加されたのを確認したら、OK ボタンか適用ボタンを押す。

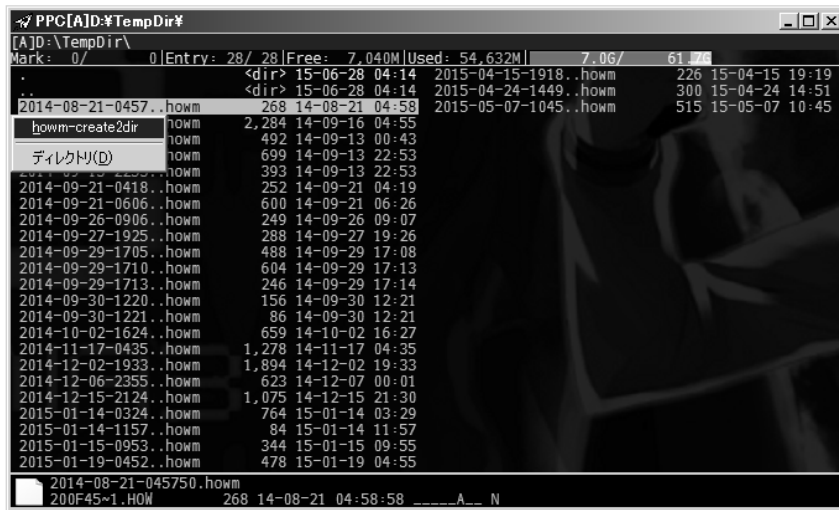


基本操作

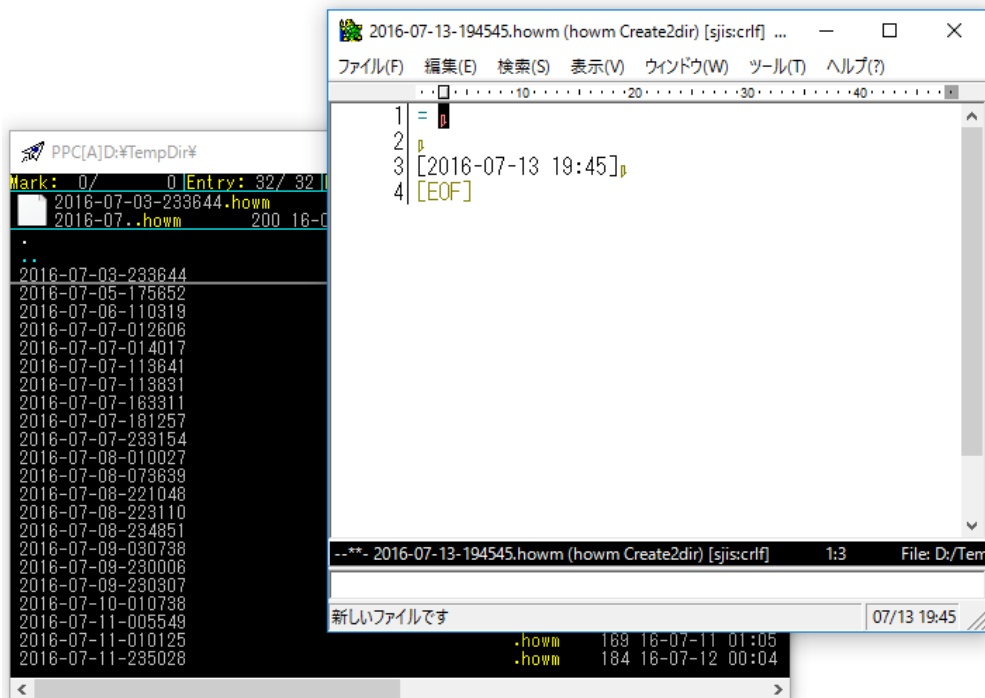
まずは、MoE の基本操作を確かめてみよう。

1. メモの作成

適当なフォルダで[Shift+K]を押すと、ファイル作成メニューが出るので、howm-create2dir を選択する。

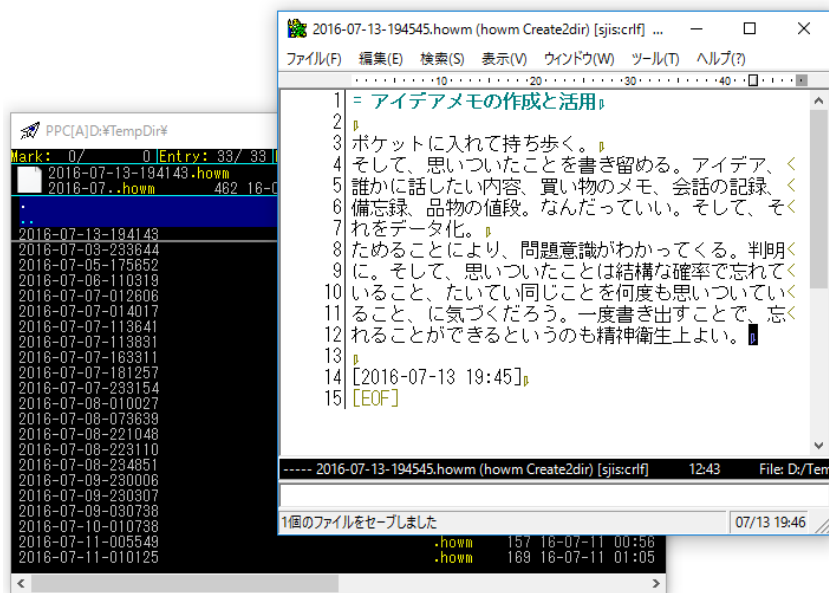


すると、howm 形式で xyzzzy が起動する。一行目にはタイトル、それ以降には本文を書く。ここで作成したファイルは、カレントフォルダに保存される。



あとは、以下のキーバインドを覚えておけばいい。ちなみに、行頭に*や**を付けるキーバインドがあるのは、見出しのため。行頭が**なら見出し、***なら小見出しとなり、行の色がそれぞれ変更される。

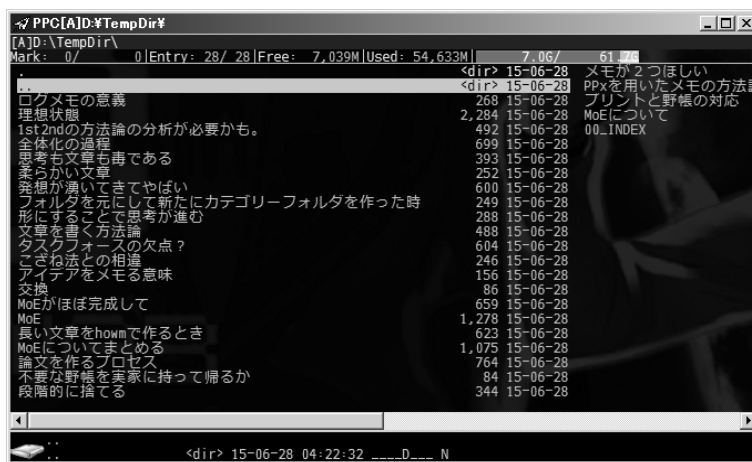
- C-c, c 新しいメモを開く (Ctrl と C を同時押ししたあと、「,」を押して、C を押す)
- C-x s メモを保存 (Ctrl と X を同時押ししたあと、S を押す)
- C-x C-c 終了 (Ctrl と X を同時押ししたあと、Ctrl と C を同時に押す)
- C-: **を行頭に付ける
- C-] *を行頭に付ける



2. タイトル一覧表示

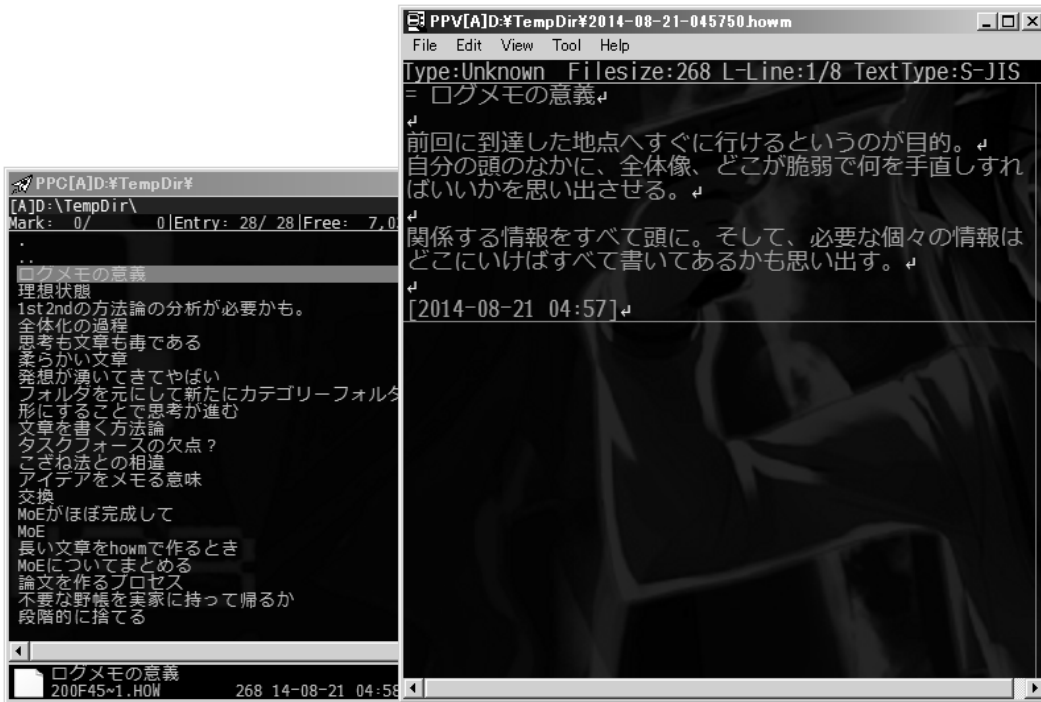
メモファイルがあるフォルダで、[F6]を押すと、メモファイルの一行目とファイル名とを対応させたコメントファイル「00_INDEX.txt」が作成される。

そののち、[;]を押して表示形式メニューを表示し、howmtitle を選択すると、タイトル表示になる。



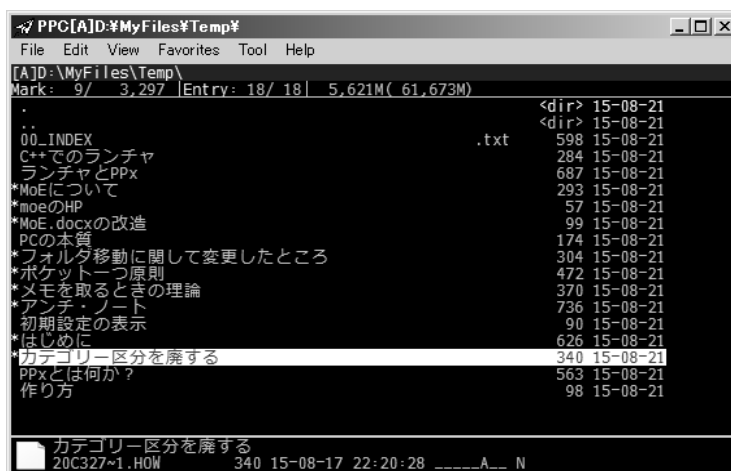
3. メモの閲覧

メモファイルにカーソルをあわせ、[Enter]を押すと、ビューアが起動する。カーソルキーで表示ファイルを切り替えることができる。もう一度[Enter]を押すと、ビューアは閉じる。



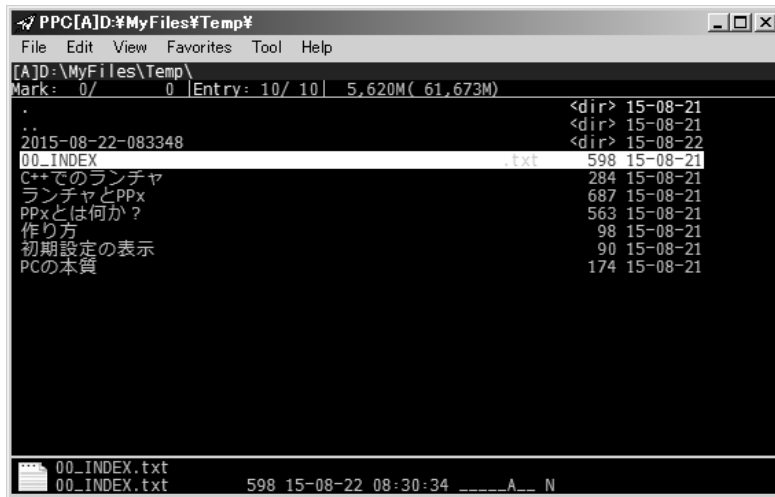
4. ファイルをマーク

[Space]を押すと、カーソル下のファイルをマークすることができる。



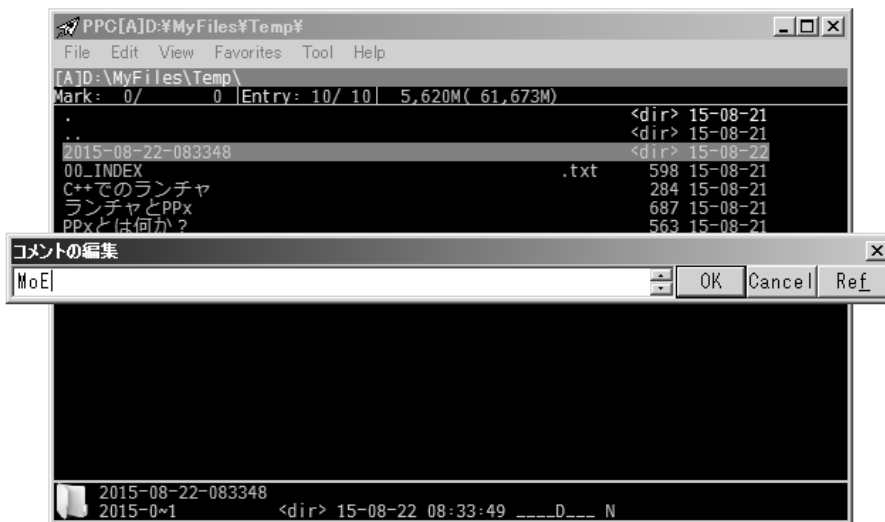
5. 複数のメモを一つのフォルダにまとめる

関係するメモをマークしたあと、[Alt+M]を押す。すると、カレントフォルダに新しくフォルダを作成し、その中にマークファイルが移動する。この時、フォルダ名はタイムスタンプから自動で付けられる。



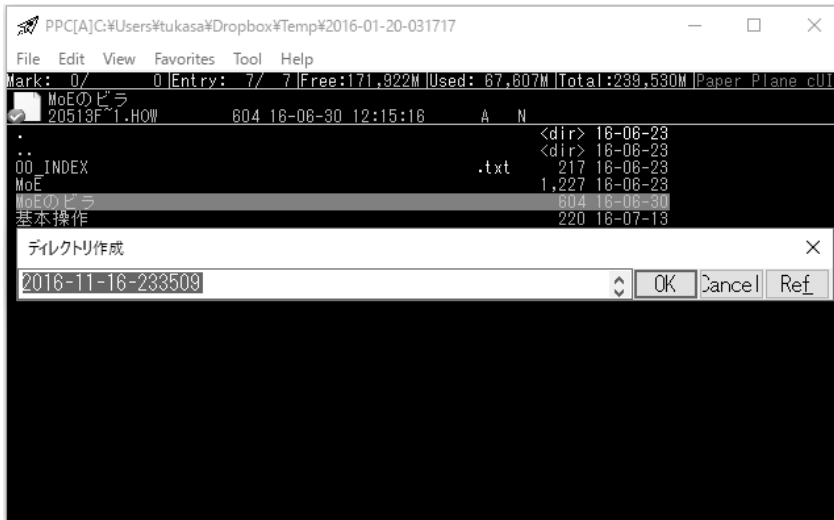
6. フォルダに名前を付ける

フォルダにカーソルをあわせ、[Ctrl+R]で名前を変更できる。この時、タイトル一覧表示と同様に、コメント機能を利用して表示名を変更しているので、同一フォルダに既にあるフォルダ名を使っても問題ない。実際のフォルダ名を変えたい場合は[R]で変更する。



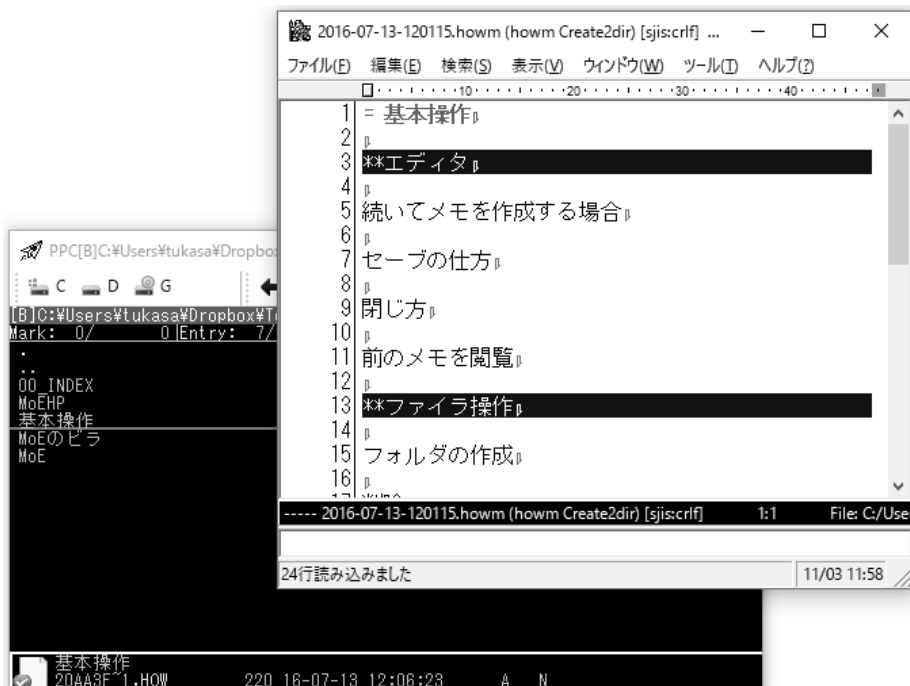
7. フォルダの作成

[K]を押す。



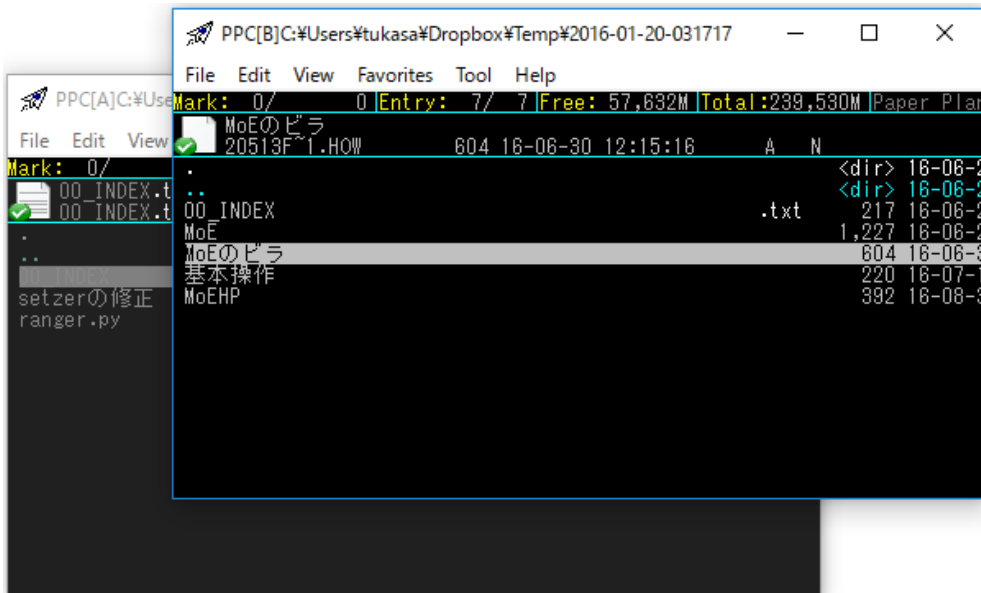
8. ファイルの編集

編集したいファイルにカーソルをあわせ、[E]を押すと編集できる。



9. 窓間移動

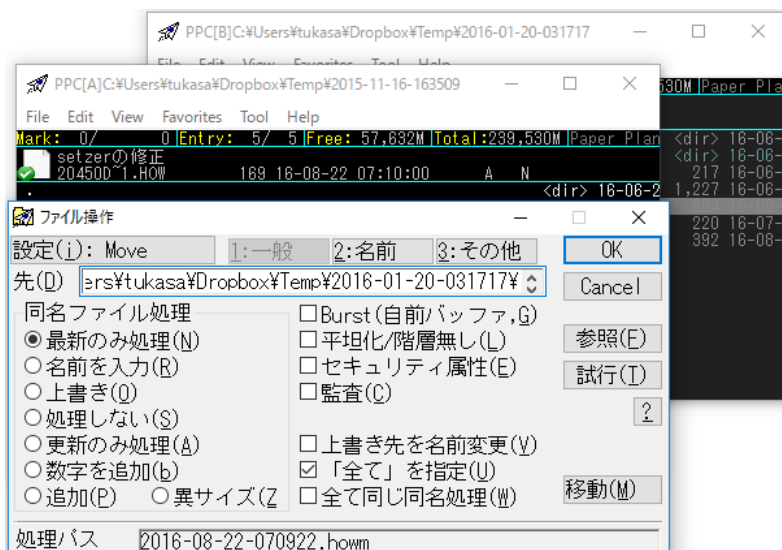
[Tab]を押すと、窓をもう一枚開く。すでに開いている場合は、窓間移動をする。



10. ファイル移動

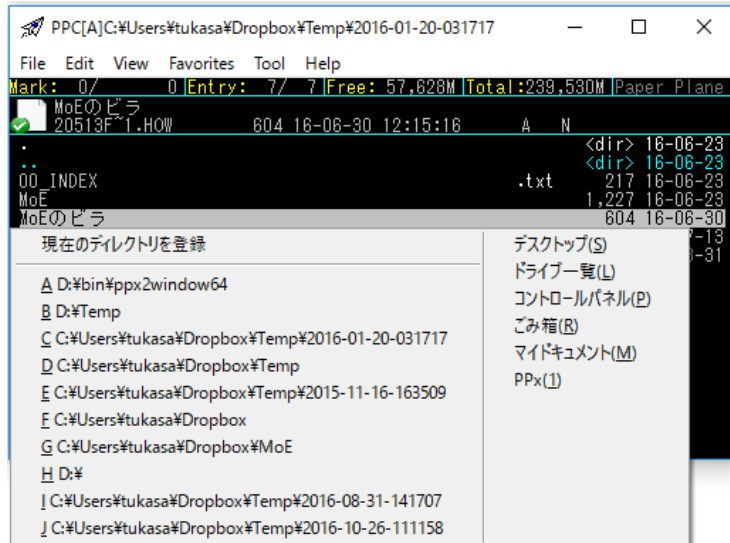
反対窓で移動先のフォルダを開いてから、現在窓で移動させたいファイルにマークをする。[M]を押すと、ファイル操作ダイアログが出る。処理先に反対窓のパスが入力されているのを確認したら、[Enter]を押す。これで、ファイルが移動する。

移動させたいファイルが一つの場合のみは、そのファイルにカーソルをあわせて[M]でいい。また、ファイルコピーは[C]でできる。



1.1. ブックマーク

[O]を押すと、最近移動したフォルダー一覧とブックマークしたフォルダー一覧が表示される。選択すると、そのフォルダに移動する。「現在のディレクトリを登録」を選択することによって、現在表示しているフォルダを一覧の末尾に追加できる。



1.2. ソート状態の変更

[S]を押すとソート状態を変更できる。名前逆順か更新日付逆順あたりを選択すると、最近作成したメモファイルが前の方に来るのでおすすめ。

1.3. その他

[BS] 親ディレクトリへ移動

[F5] 表示ディレクトリの再読み込み

[Q] 終了

運用

では、実際にメモを運用する段階へと移ろう。まずは、2つのフォルダを適当な場所に作成する。

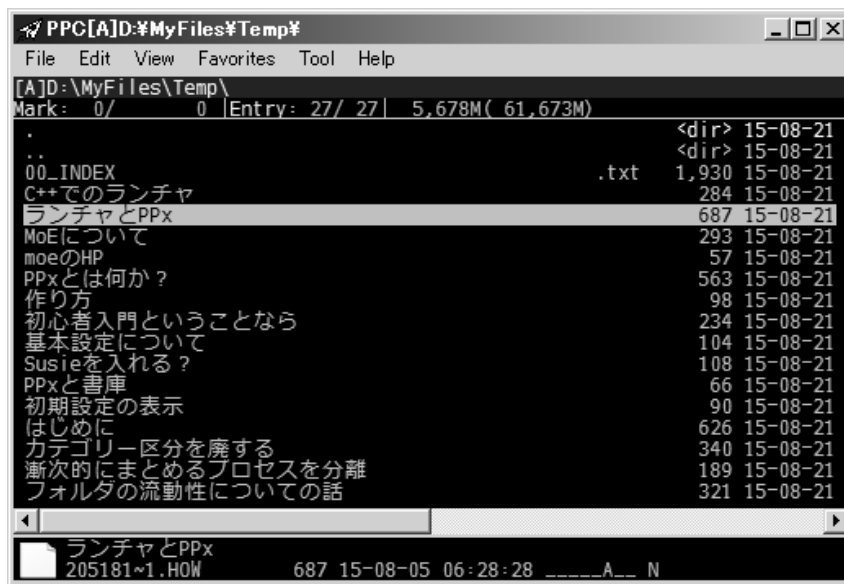
- 一時フォルダ (Temp) ……一時的にメモをためるためのフォルダ
- ゴミ箱フォルダ (Dust) ……不要なメモを捨てるためのフォルダ

PPx では、[K]を押せばフォルダを作成することができる。

1. メモをためる

何か思いついたことがあれば一時フォルダに行き、[Shift+K]からメモファイルを作成する。

この時、カテゴリー分けをしようなどと考えるはいけない。どのようなメモであっても、まずは一時フォルダに、一切の区別なく保存するのだ。メモの種類も、カテゴリーも、以前に書いたメモとの関係性も気にせず、どんどん思いつくままにメモをためていこう。



メモを作る際のポイントは、とりあえず書く、ということ優先し、完成度を無視することである。内容が曖昧でもいいし、文章として成り立っていない、片言のものであってもいい。とりあえず、書くのだ。また、その際には既存のメモの内容を全く考慮しないで書く。もしかしたら以前、同じ内容のものを書いているかもしれないし、かつて書いたものと、今書いているものの内容は、矛盾しているかもしれない。それでもいいから、とりあえず書くのである。整合性に関する考慮は、メモをどんどん作ることの妨げになるのだ。

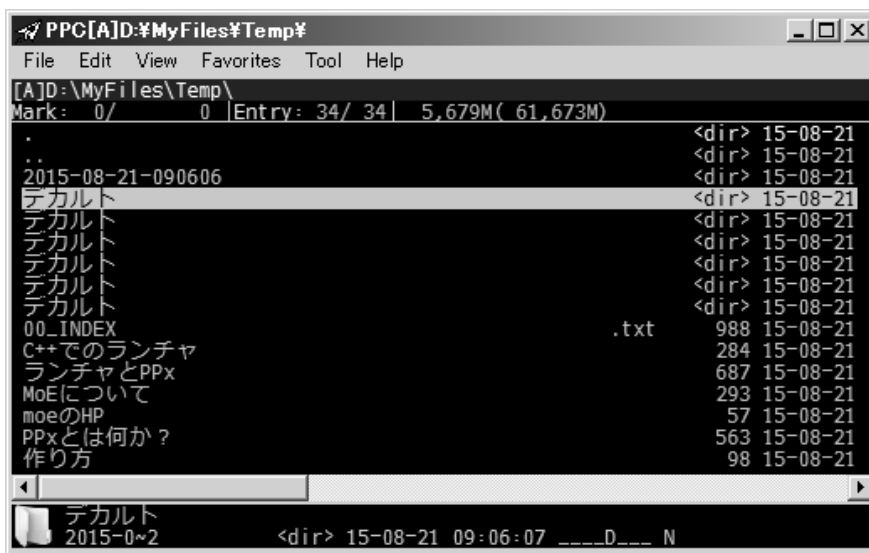
2. メモをカテゴリー分けする

ある程度時間が経過すれば、各々のメモが、どのような問題意識で書かれたものなのかが明確になってくる。そうしたら、そのとき初めてカテゴリー分けを行う。

一時フォルダに行き、タイトル表示とビューアで内容を確認しながら、同じカテゴリーのメモファイルを[Space]でマークしていく。マークが終わったら、[Alt+M]を押す。すると、カレントフォルダにフォルダが自動で作成され、その中にマークしたファイルが移動する。こうやって、一時フォルダ内に個々のカテゴリーフォルダを作っていく。

既にカテゴリーフォルダが存在する場合には、そこにメモファイルを移動するのもいいだろう。

新しく作成したフォルダには、わかりやすい名前を付ける。フォルダにカーソルをあわせ、[Ctrl+R]を押して適当な名前を入力しよう。この際はコメント機能を利用しており、実際のフォルダ名を変えているわけではないので、同一フォルダ内に既に存在するフォルダ名を付けることも可能だ。



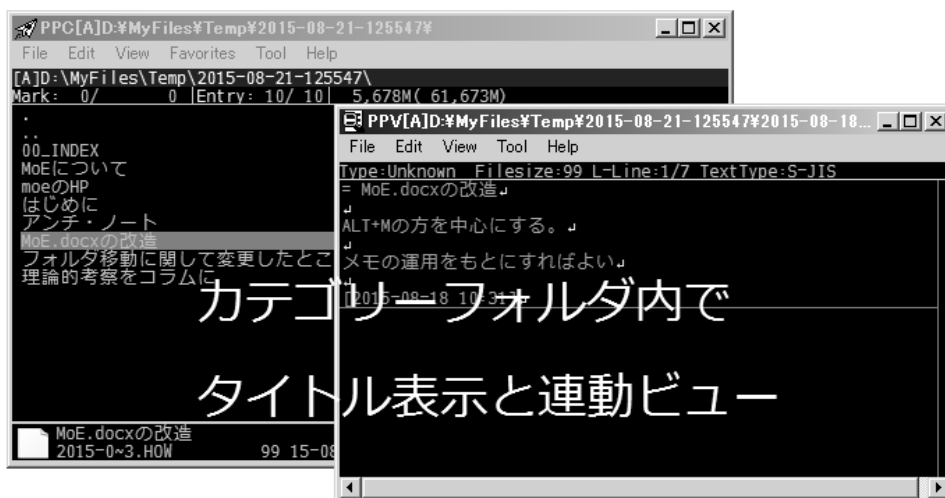
以上の作業と平行して、メモの削除も行う。作成済みの文書に内容が反映されているメモや、状況の変化によって問題意識自体が古くなっているメモを見つけたら、その都度、ゴミ箱フォルダに移動する。[Tab]で反対窓を開き、ゴミ箱フォルダを表示してから、現在窓で不要なファイルを[Space]でマークして、[M]で移動だ。

3. メモをまとめる

カテゴリー分けが終わったら、個々のカテゴリーフォルダに入り考察を行う。基本的には、そこにあるメモファイル同士を見比べながら、一つのファイルにまとめる作業になる。断片的なアイデアを、全体的でまとまりのあるものにしていくのだ。

まとめる際は、メモファイルの一つをエディタで開き、全体構成を意識しながら、他のメモファイルの内容をコピーするのが簡単だ。新たにメモファイルを作成し、既存のメモファイルの内容を参考にして文章を作る、というのもいい。そうしてメモファイルを統合したら、利用済みのメモファイルはゴミ箱フォルダに移動する。

まとめる過程で新たに思いついたことがあれば、それもメモファイルにする。一時フォルダに作成してからカテゴリ分けをするのでいいし、直接カテゴリフォルダ内に作るのもいい。時間が経過したら、またそれを他のメモファイルと統合するのである。カテゴリフォルダ内のメモファイルの増減を経て、メモファイルは徐々に、精度の高いものになっていくのだ。



フォルダ名の考察

日時から自動でフォルダ名をつけ、コメントで表示名を変更することには、いくつか利点がある。その一つが、あるメモ郡をまとめる時に、一々そのまとまりに名前を付ける手間を省けることである。例えば「さっき考えたこと」「この論文のこの箇所について考えたこと」といったように、明確にカテゴリ名を決められないものに対してでも、カテゴリ分けが可能になるのである。このことにより、メモをカテゴリズするための精神的なハードルを下げるのだ。

また、メモのまとまりに名前を付ける際に、そのフォルダ名が他のフォルダ名とかぶっていないかを一々気にかける必要が無くなる、というのも大きい。

なぜ削除ではなくフォルダ移動か？

ここでは、メモを捨てるとは言っても、実際に捨てているわけではない。別のフォルダに移動して、見えなくしているだけである。

ファイル削除ではなく、ファイル移動で済ませているのは、捨てるハードルを低くするためである。ファイル移動であれば、ゴミ箱フォルダに捨てたとしても、あとで頑張れば、Grep なり何なりを使って取り戻すことができる。つまり、不要なメモを捨てる際に、一々これは本当に捨てていいのか、後で後悔しないか、という究極の決断をする必要がなくなるのである。こうして、削除のハードルを下げることで、一時フォルダ内とカテゴリフォルダ内の秩序を維持しやすくしているのである。

タイトルはファイル名で実装するか、それとも一行目で実装するか

テキストファイルでタイトルを実装する方法は、二通りある。

- ファイル名をタイトルにする
- テキストファイルの一行目をタイトルにする

ファイル名をタイトルにする、というのはそのままだ。たとえば、「MoEーテキストメモの考察」というタイトルにしたいのなら、ファイル名を"MoEーテキストメモの考察.txt"とする。

テキストファイルの一行目をタイトルにする場合は、ファイル名はタイムスタンプから自動的に付けるようにする。そしてタイトル表示は、ソフト側の機能を用いて行う。howmがこの形式だ。

前者は、メモを作る用途においては不適當である。メモをどんどん作成するには、余計なことを意識せずに済む状況が必要なのだが、それに反するからだ。理由を順に見ていこう。

同一タイトルにできない

同じフォルダ内に、同一タイトルのメモを複数作れないことがネックになる。ファイル名が重複してしまうからだ。そこで、一々ファイルを作るときに、フォルダ内に同じタイトルのファイルがないかどうかを一々意識しなければならなくなる。

タイトルが最初から決まっているわけではない

メモのタイトルは、常に最初から決まっているわけではない。とりあえず本文を書き出してみて、自分が何を書こうとしているかがはっきりしてからタイトルを付けることもあれば、最初に付けたタイトルを書いている途中で変える、ということもあるわけだ。タイトルは、好きなときに付けられて、好きなときに変更できる方がいい。だが、ファイル名をタイトルにしてしまうと、その融通がきかない。だから、一行目をタイトルにするほうが適切なのである。

ツール

アナログツール

野帳とらくがき帳

常に PC の前に座っている人でも無い限り、手帳なり何なりのアナログツールに思いついたことを書き付けて、あとでそれをデータ化をする、ということになるだろう。

アナログツールに必要な要素は、

- 方眼あるいは白紙
- バラバラにならない
- 安価
- 持ち運びがしやすい

あたりである。あとでデータ入力をするを考えると、アナログツールは一つに集約しておいた方がいい（ポケット一つ原則と言う）。そこらへんにある紙切れに適当にメモをしたとしても、あとでそれを紛失する危険性があるからだ。また、適当なアイデアでも気兼ねなく書けるように、安価であるほうがいい。

私は現在、ココヨの測量野帳を利用している。一冊 40 枚（80 ページ）あり、200 円以下で購入可能である。新書の丈を少し短くしたくらいの大きさなので、ポケットに入れて持ち歩くことも容易だ。また、カバーが硬いので、外で立ったままメモをすることも可能だ。他の選択肢としては、無印良品のらくがき帳がある。携帯性は野帳に劣るが、コンビニで買えるためにストックが必要ないという利点がある。価格も 100 円以下で安い。また、不要になった紙面を破り捨てるのが容易なため、野帳のようにデータ化して用済みになったものがたまる、という問題もない。

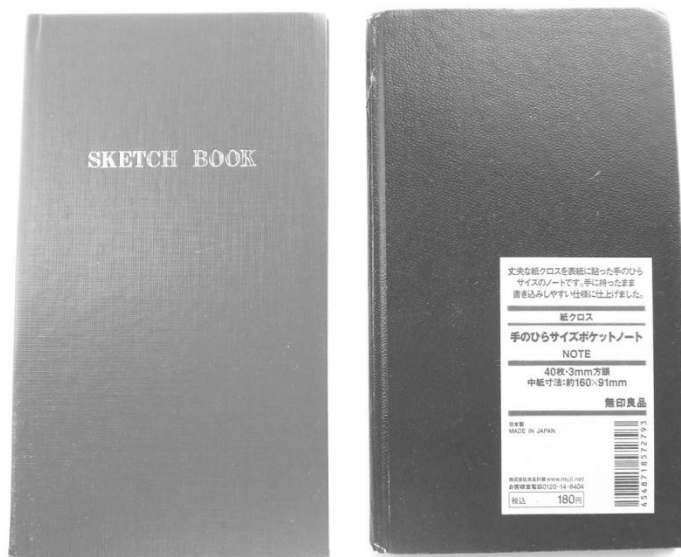
メモと編集の分離

手帳でデータベースをつくらうという試みは、基本的に失敗すると考えていい。というのは、メモをする際に求められることと、後でそれを利用する際に必要になることが原理的に矛盾しており、個々の工夫で克服できるようなことではないからだ。

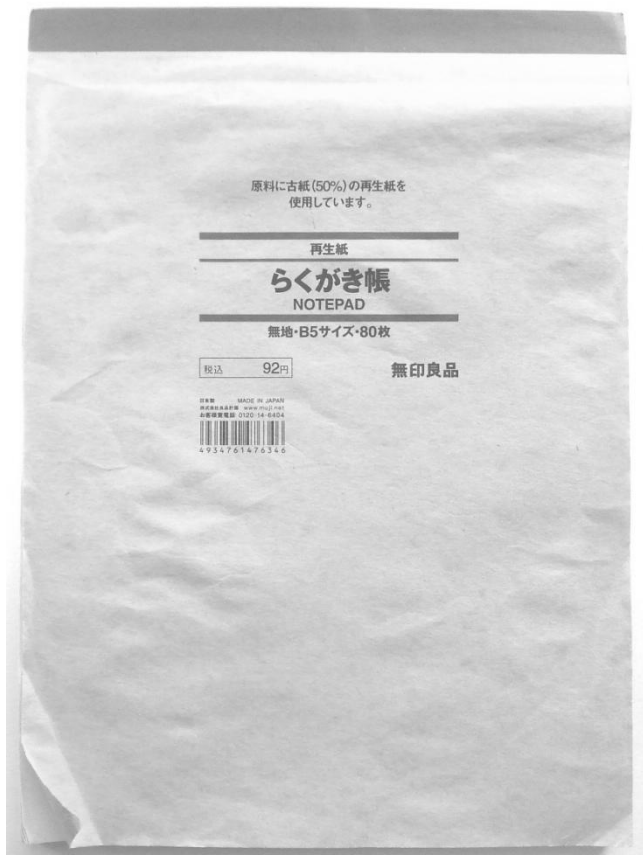
前者で必要なのは、何も考えずに書けることだ。他のメモとの関係性だとか、以前に何を書いたかなどは一切無視するべきであり、時系列で書くことが望ましい。後者で必要なのは、カテゴリズであり、時系列とはまた別の秩序を必要とする。この両者は本質的に矛盾するのだ。だから、この二つは物理的に分けた方がいい。そこで、まずアナログツールによって時系列順にメモをとってから、あとでそれをデータ化する、という段階的な処理をしているのである。

手帳が一冊程度であれば、目次をつける等のちょっとした工夫によって、この問題は何とかなるかもしれない。しかし、時間の経過とともに扱うべきメモは増大していく。だから、この手の試みは、必ずどこかで破綻することになるのだ。

左が野帳。右はファミマに売っている無印良品の手のひらサイズポケットノートで、中身はほぼ野帳と同じ



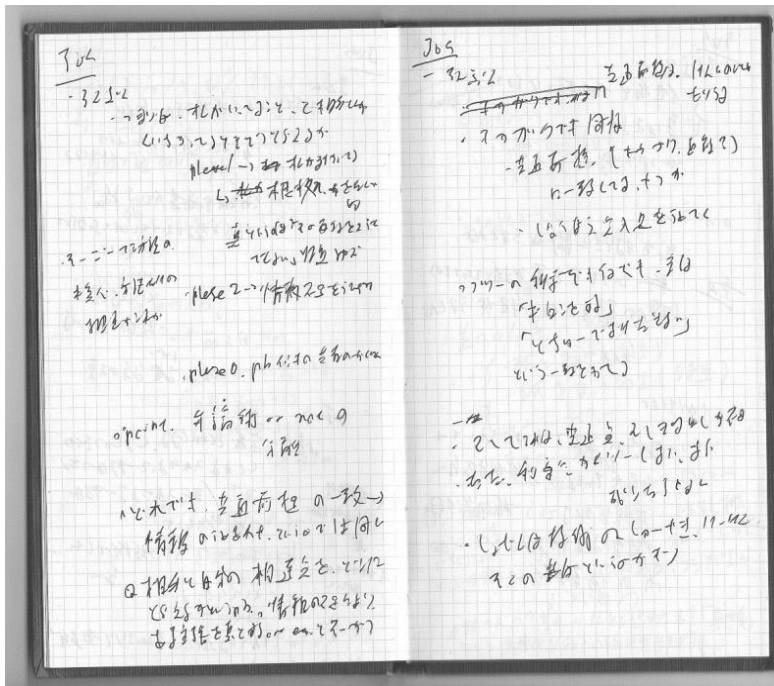
無印良品のらくがき帳。同じくファミマで買える



メモの書式

アイデアは、思いついた時間順にびっしりと、詰めて書いていく。京大式カードのように、一枚一項目原則にはこだわらない、ということである。もちろん、こうすると同じカテゴリーのメモが飛び飛びに現れることになる。だが、それはデータ化したあとに解消されるのだから、気にする必要は無いのである。また、文字は汚くても問題ない。どうせすぐにデータ化するからだ。書いてからデータ化するまでの期間、自分が読めればそれで十分なのである。書く際の書式は自由でいいが、日付だけは入れるべきである。メモ全般に言えることだが、日付の無いメモは使い物にならないことが多い。また、データ化の際の参考になるよう、簡単な見出しをつけてもいい。

野帳の使用例



常に持ち歩く

アイデアを書き留めるための野帳とボールペンは常に持ち歩くようにする。かばんがあるならそれに入れてもいいし、ポケットに入れて持ち歩いてもいいだろう。そして、思いついたことや気になったことはすべてそこに書き留める。新しいアイデアでもいいし、誰かに話したい内容でも、会話の記録でも、買う必要のある商品名でも、何でもいい。とにかく区別せずに書き留めよう。

デジタルツール

ノート PC やポメラといった、持ち歩けるツールがあれば、出先であってもメモのデータ化ができて便利である。自分の家に戻った後に、野帳に記したメモを一気に打ち込むという作業は、しばしば億劫になるからだ。この際、DropBox などを使い、複数の PC 間でメモファイルを同期させると便利である。

記録メモ

アイデアメモは基本使い捨てだが、それ自体が価値を持ち、後々まで残して置いた方がいいメモがある。日記、誰かとの会話記録、会議記録、本の感想、などといった記録に関するものである。

この種のメモは、本来的にアイデアメモとは異なるものであるが、最初は一時フォルダに入れることになる。メモを作成するときに、一々それが記録メモかどうかを判別するのは手間だからだ。その後のカテゴリー分けの際に、記録メモを、その他のアイデアメモとは分けて管理するのがいい。私は、記録メモ保存用フォルダを別に作って、そこに移動するようにしている。

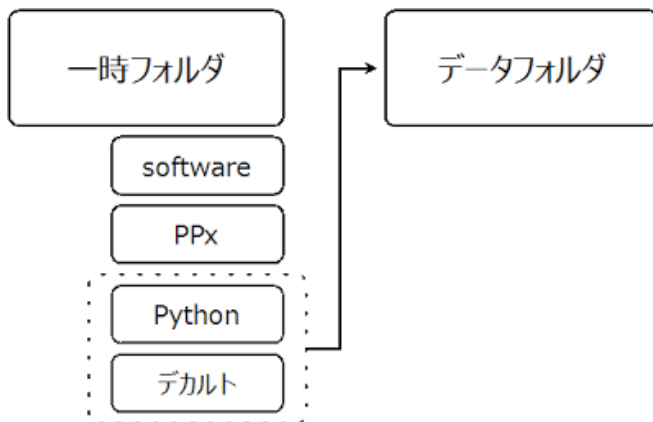
変化の無いメモを別フォルダへ

変化の無いメモ

時間が経過するに従い、カテゴリーフォルダの中に、長期間変化も無いまま居座っているものが出てくるだろう。数ヶ月間、その中にあるファイルが増減したことも、編集されたこともない。それどころかほとんど参照すらしていないようなフォルダである。

かつて、眼前にある問題を解消しようとして、そこにまとめられているメモを作ったかもしれない。だが、状況の変化によって、実際にはその問題を解決してしまったわけではなくても、当面は問題にしなくてもよくなることもある。そのような場合、このような形でフォルダが取り残されてしまうわけだ。

そのようなフォルダを、一時フォルダ内にずっと置いておくのは鬱陶しい。現在の問題意識から外れたフォルダが存在することは、思考のノイズになるからである。一時フォルダには、今取り組んでいる課題に関するファイルとフォルダだけが存在してほしいのだ。しかし、だからといってそれをゴミ箱フォルダに捨ててしまうわけにもいかない。特にその問題意識が解決したというわけではないので、後々必要になるかもしれないからである。



データフォルダに移動

これを解決するため、新たにフォルダ（ここではデータフォルダと呼ぶ）を適当な場所に作成する。

そして、ほとんど変化の無くなったメモファイルは、フォルダごとデータフォルダに移動する。こうすることで、一時フォルダ内に使っていないフォルダがあって気が散る、ということも無くなるわけだ。

データフォルダについて

一時フォルダからデータフォルダに移動する際は、内容が似通ったフォルダを一つのフォルダにまとめよう。細かい区分は後々無意味になるからだ。一時フォルダでは、同じタイトルのカテゴリーフォルダがいくつも並んでいて問題なかった。今、取り組んでいる課題であれば、細かな区別をしていたとしてもそれを把握できるからである。しかし、データフォルダの場合は、その前提が違ってくるのである。ま

た、それゆえ、一時フォルダでそうであったように「フォルダ名はタイムスタンプで付けてコメントで区別する」という方式を取る必然性もない。フォルダ名自体を変更してもいいだろう。

データフォルダ内にあるカテゴリーフォルダの扱い自体は、一時フォルダ内のものと同様でいい。タイトル表示と連動ビューで活用し、不要なメモを見つけたら適宜ゴミ箱フォルダに移動しよう。

論文メモ

特定の期日までに、論文やレジュメといった文書をまとめる必要がある場合がある。このような時は、意識的にアイデアを出して、それを短期間の内にまとめることになる。

大雑把な文書を作る

まず、喫茶店などに行って、短期間に集中して、アイデアを野帳からくがき帳に書き出す。この時は、とにかく頭にあることをすべて書き出すことを心がける。

このメモをデータ化し、それを元にしてとにかく一つ、文書をつくってしまう。アイデアメモを作ってから、文書にするまでの期間は、できるだけ短い方がいい。せいぜい一週間程度だろうか。そうしたら、それを徐々に修正する形で、完成に近づけていく。

完成した文書を一度に作るのでは無く、荒削りでもとりあえず形にし、それを徐々に修正して完成に近づける、というイメージである。一度文書を作ったならば、自分の理論のどの点が不十分であり、何について知らなくて、何を新たに考察する必要があるのかが見えてくる。そして、次はそれを念頭に置いた上で考察し、またそれをメモの形で書き出し、それを元にして、文書を修正する。それでまた不十分な点が見つかれば、またその箇所を修正する……というように、漸次的に文書を完成させるのである。そうやって全体の形を整えていき、最後に文体や誤字や表現といった、細かいところに手を入れる。

そのような方針が無いところで、いくら考えたとしても、それは結果に結びつかないことが多い。全体的な指針が無いと、それが文書の完成に寄与するかどうかと無関係に、その時たまたま目についたことを、闇雲に考察することになってしまうからである。労力としても時間としても、無駄なのだ。準備を万全にして、個々の論証の流れや全体の論理展開を頭に思い描き、いきなり完成した文書を作り出す必要性など無い。最初はどれだけ適当なものであってもいいのである。

紙媒体の利用

文書の修正をする際は、PC 上だけでやるのではなく、紙媒体も併用するのがおすすめである。そのほうが、全体の構造を把握しやすいのだ。

プリントアウトしたら、それを読みながら、ペンで書き込みを加えていく。それが終わったら、その書き込みを元のデータに反映する。そうしたらまた、そのデータをプリントアウト。また書き込み、反映…という過程を繰り返すのだ。

印刷例。xyzyy の印刷設定で二段組にしている



文書の分量が増えたら

文書が数千字以上の分量になると、見出しの階層化や章分けが必要になり、テキストファイルでは対応できなくなる。そのときには、文書を Word や Tex といった形式で作ることになる。

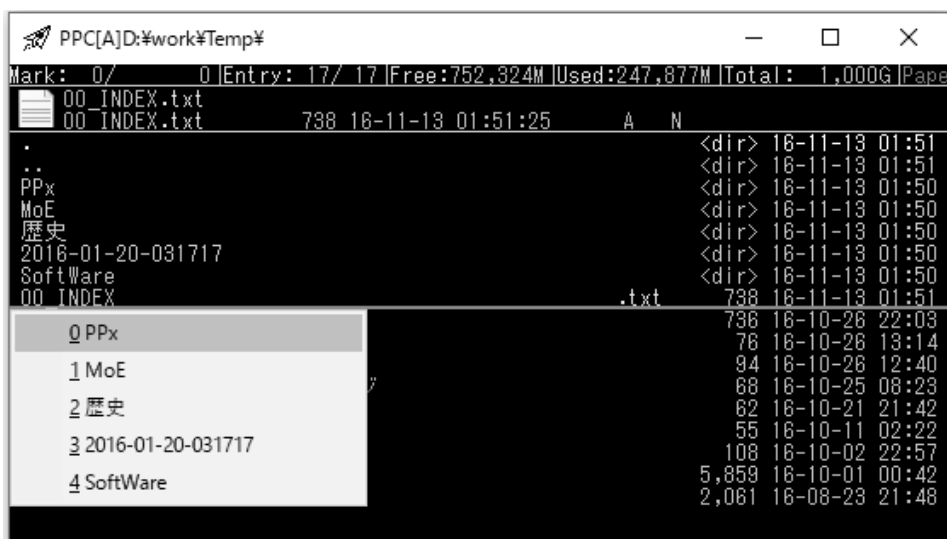
ここでもやることは基本的に同じである。文書をプリントアウトし、そこに書き込みをする形で発展させていく。そしてその過程で思いついたことがあれば、それをメモファイルにし、後々そのまとめ文書に統合する。そしてまたプリントアウトして……という過程を繰り返し、徐々に文書を洗練していく。

応用編

MoE をさらに便利に行うための方法をいくつか紹介する。PPx についてより詳しく知り、自分なりのカスタマイズをしたいという場合は、私が PPx のカスタマイズについて解説しているサイト [ppxmemo](https://sites.google.com/site/ppxmemo/) (<https://sites.google.com/site/ppxmemo/>) 等を参考にして欲しい。

1. サブフォルダー一覧をメニュー表示してファイル移動

サブフォルダー一覧をメニュー表示し、そこにマークファイル（あるいはカーソル下のファイル）を移動する。メモファイルを既存のカテゴリフォルダに移動する時に便利。



PPx の Script フォルダに、次のファイルを保存する。

subdirmenu.js

以下のコードをクリップボードにコピーし、「編集して取込」をする（24 ページ参照）。

```
KC_main = { ; PPc メイン窓
^M,*script %0\Script\subdirmenu.js,m
}
```

移動したいファイルをマークし（移動させたいファイルが一つだけならカーソルを置くだけでもいい）、[Ctrl+Shift+M]を押すとサブフォルダー一覧メニューが表示されるので、移動先のフォルダを選択する。

2. 任意のフォルダから一時フォルダにメモを作成

任意のフォルダから一時フォルダにメモを作成できるようにする。

以下のコードをクリップボードにコピーし、「編集して取込」をする。

```
KC_main = { ; PPc メイン窓
^M,%Ob,xyzzzy -e (howm-create2dir) C:\Users\tukasa\Dropbox\Temp
}
```

パス (C:\Users\tukasa\Dropbox\Temp) は各自の一時フォルダのパスに変えよう。

これで、[Ctrl+M]で、メモを一時フォルダに作成することができる。

3. [D]でメモファイルをゴミ箱フォルダへ移動

以下のコードをクリップボードにコピーし、「編集して取込」をする。

```
KC_main    = {      ; PPc メイン窓
D ,*ifmatch      "/C:\\Users\\tukasa\\Dropbox\\Temp/",%1          %:
*ppcfile        !move,C:\Users\tukasa\Dropbox\Dust,/sameall:on      /same:5
/querycreatedirectory:off %: *stop
                %K"@D"
}
```

最初のパス (C:\Users\tukasa\Dropbox\Temp) は各自の一時フォルダのパスに。二つ目のパス (C:\Users\tukasa\Dropbox\Dust) は各自のゴミ箱フォルダのパスに変えよう。

これで、一時フォルダ内にいるときに [D]を押せば、マークファイル (あるいはカーソル下のファイル) をゴミ箱フォルダに移動することができる。

参考文献・リンク

参考文献

梅棹忠夫『知的生産の技術』1969
野口悠紀雄『「超」整理法』1993
野口悠紀雄『続「超」整理法・時間編』1995
野口悠紀雄『「超」整理法〈3〉』1999

サイト

TORO's Library (<http://toro.d.dooo.jp/index.html>)
github 版 xyzyy 配布サイト (<http://xyzyy-022.github.io/>)
PoIC (<http://pileofindexcards.org>)

著者のサイト

Memo on the Electron (<https://sites.google.com/site/moesystem/>)
つかさのほえほえ日記 (<http://hoehoetukasa.blogspot.jp/>)
ppxmemo (<https://sites.google.com/site/ppxmemo/>)

Memo on the Electron ver 2.0 発行日 : 2016 年 11 月 19 日 発行者 : 柗つかさ Mail : kihuyutukasa@gmail.com Twitter : @tukasa1900
--