

2012 年 04 月 13 日

計算機実習 IV (2012 年度)

第 1 回: コマンドプロンプトとバッチファイル その 1

目次

1	コマンドプロンプトの使い方	1
2	otbedit の使い方	4
3	バッチファイルの実行	5
4	バッチファイルの変数	6
5	バッチファイルのオプション	8
	コラム: MS-Windows のスクリプト環境	8

1 コマンドプロンプトの使い方

コマンドプロンプトの使い方について簡単に紹介する。

- 起動

[スタート] ⇒ [プログラミング] ⇒ [計算機実習 IV]¹

- プロンプト

コマンドプロンプトでは、デフォルトでは

```
C:¥Documents and Settings¥hoge¥My Documents> dir
```

のように、プロンプト記号 > の左に、今自分がいるディレクトリ (カレントディレクトリ) が表示され、ユーザが入力するのは > の右の「dir」の部分であるが、以下では、必要のある場合以外はカレントディレクトリ部分は省略して

```
Z:> dir
```

のように書くことにする。

¹計算機実習 4 用のコマンドプロンプトは、普通のコマンドプロンプトに名前をつけて、環境変数などをいくつか追加したもの。

- コマンド入力

コマンドプロンプトでは、> の右側に「コマンド名」を入力し、Enter キーを打つことでそのコマンドを実行する。コマンド名の後ろにスペースを空けてオプションを続けて書くこともある。コマンド名は、大文字でも小文字でもよい²。

コマンドの実行方法について、注意すべき点をいくつかあげる。

1. 拡張子の有無

実行ファイル名 (拡張子 .exe) はそれ自身コマンド名となるが、その場合拡張子は必要ない。実行ファイル以外でもファイル名 (正確には拡張子名) に関連づけされたアプリケーションがある場合、コマンドプロンプトでそのファイル名をコマンド名のように入力すると³、そのアプリケーションを起動してそのファイルを開いてくれる。

2. ヘルプ

コマンドの説明 (ヘルプメッセージ) は、例えば dir という名前のコマンドであれば

```
Z:> help dir
Z:> dir /?
```

のように help コマンドを使うか、/? オプションを指定すれば、コマンドプロンプト画面にヘルプが表示される⁴。

3. 履歴

コマンドプロンプトに過去に入力したコマンドは、上下の矢印キーで履歴を呼び出すことができるので、再び同じ行を入力せずに済む。左右の矢印キーでカーソルを動かして入力を修正できるが、その際、カーソルの現在位置が行の中間であっても、Enter を入力すれば行全体がコマンドとして渡される⁵。

4. TAB 補完

コマンドプロンプトでファイル名を入力する場合、先頭の何文字かを入力した後に TAB キーを入力すれば、それで始まるファイル名を補完してくれる。該当するファイルが複数ある場合は、そのまま TAB キーを繰り返し打つことで該当するファイル名を巡回してくれる。

コマンドプロンプトで使える簡単なコマンドを以下にあげる。

²コマンド名やファイル名の大文字小文字は、コマンドプロンプトでは区別されない。オプションは大文字小文字の区別がないものも多いが、例外もある。

³この場合は拡張子も必要。

⁴MS-Windows に用意されている標準的なコマンドの場合。

⁵ワープロの改行とは違いカーソル位置の左の部分だけが入力されるわけではない。

- dir : ディレクトリリストの表示
- cls : コマンドプロンプト画面のクリア
- help : コマンドの説明の表示
- date /t : 今日の日付の表示
- time /t : 現在の時刻の表示
- start [コマンド]
[コマンド] を別ウィンドウで実行⁶。なお、[コマンド] として URL や mailto:[メールアドレス] やファイルを指定すると、それぞれ、規定のブラウザ、規定のメーラ、ファイルに関連づけられたアプリケーションを別ウィンドウで起動してそれらを開く。
- echo [文字列]
[文字列] をコマンドプロンプト画面に表示する。ただし [文字列] が ON か OFF (小文字でもよい) という文字列の場合、コマンド行自体の表示を ON または OFF にする⁷。指定する文字列は " " のような引用符では囲まない。引用符をつけるとそれも文字列に含まれる。
- color [色指定]
コマンドプロンプト画面の色指定。[色指定] は 2 桁の 16 進数で上の桁が背景色、下の桁が文字色。色は、0~7 は暗色の黒、青、緑、水色、赤、紫、黄、白、8~f が明色の黒、青、緑、水色、赤、紫、黄、白。オプションをつけずに実行するとデフォルトに戻す⁸。

最後の color は、例えば

```
Z:> color 6a
```

とすると背景が暗色の黄色で、文字色が明色の緑になる。

課題 1-1. コマンドプロンプトを立ち上げよ。

課題 1-2. コマンドプロンプトの背景色、文字色を変更し、使いやすい色を選択せよ。

⁶ 囲んでいる [] は、オプションであることをわかるように書いているだけなので、[] は実際にはつけないことに注意。今後もオプションなどをこのように [] で囲んで書くことがあるが、特に補足しない限り [] はつけない。

⁷ ON, OFF の機能は主にバッチファイルで使用する。

⁸ コマンドプロンプトの [プロパティ] でより細かく設定することも可能。

課題 1-3. コマンドプロンプトで以下のコマンドを実行し、結果を確認せよ。

- dir
- cls
- help color (color /? でもよい)
- notepad
- cmd
- start <http://www.niit.ac.jp>

なお最後のものは、実行前に先にブラウザを立ち上げておかないとうまくいかないかもしれない。

課題 1-4. コマンドプロンプトの履歴機能、TAB 補完機能を使ってみよ。

2 otbedit の使い方

otbedit はタブ機能付きの小さなテキストエディタで、以下で公開されているフリーソフト。プログラムソースのキーワードの色付け/カスタマイズ機能、改行文字の表示、フォントの設定などの機能がついている⁹。

- OTBEdit (A.Ogawa)
http://www.hi-ho.ne.jp/a_ogawa/otbedit/index.htm

otbedit のコマンド名は otbedit である

使い方の詳しい説明は必要ないだろうが、Ctrl-N (新規ファイルのオープン)、Ctrl-O (ファイルを開く)、Ctrl-S (ファイルを保存) のショートカットくらいは覚えておくとよい。

課題 1-5. コマンドプロンプトから otbedit を立ち上げよ。

課題 1-6. otbedit 上で Ctrl-N により複数のファイル (タブ) を開き、それを切り替えて操作を確認せよ。

⁹ 「メモ帳」や BCPad 等の他のテキストエディタでもバッチファイルは作成できるが、キーワードの色付け機能がなかったり、本講義のバッチファイルや AWK スクリプトの色付けには対応していなかったりするのので、これを利用するといいたろう。

課題 1-7. otbedit で `rem`, `echo`, `if` 等のバッチファイル用のキーワードを書き、上にある `Bat` というアイコンでバッチファイルモードにすることでそのキーワードが色づけされることを確認せよ。

3 バッチファイルの実行

バッチファイルは、基本的にはコマンドプロンプトで実行できるコマンドを、一行ずつ並べて書いたファイルである (拡張子 `.bat`)。このファイルは直接実行させることができ、実行させると一行ずつ順にコマンドを実行してくれるが、分岐、ジャンプ、ループなどの簡単な構文も備わっている。

例えば以下のように書いたバッチファイルを実行すると、ディレクトリリストを表示し (`dir`)、日付 (`date /t`) と時刻 (`time /t`) を表示する。

```
@echo off
rem バッチファイルの簡単なサンプル
dir
date /t
time /t
```

先頭の「`@echo off`」は、コマンド自体の画面表示を消すためのもの¹⁰。これを入れないと、バッチファイルに書かれているコマンド自体もコマンドプロンプト画面に表示される。また、「`rem`」(の後ろに空白つき) で始まる行はコメント行と扱われる¹¹。

バッチファイル (例えば `test1.bat` という名前のバッチファイル) を実行させるには、

1. エクスプローラー (ファイル一覧を見るもの) 上でダブルクリックする
2. コマンドプロンプト上でバッチファイル名を入力する (`.bat` は不要)

のいずれかがあがるが、前者の場合は新たにコマンドプロンプトウィンドウが立ち上がって実行され、終了したら直ちにコマンドプロンプトウィンドウが閉じられてしまうので、`pause` コマンド (後で紹介) などを入れないと何が行われたかわからないことになる。

¹⁰ コマンド名の先頭に `@` をつけると、そのコマンド自体の表示はしないので 1 行目の表示がなくなり、`echo off` によって 2 行目以降のコマンド自体表示がなくなる。

¹¹ `rem` は多分 `remark` (注釈) の略。

課題 1-8. 「ようこそ。」とだけ出力するバッチファイル `kadai1-8.bat` を作成せよ。

課題 1-9. `kadai1-8.bat` の先頭の `@echo off` を `rem` でコメントアウトして (`@echo off` を入れてなければ、逆にそれを先頭行に入れて) その動作の違いを確認せよ。

課題 1-10. 今日の日付と現在の時刻 (だけ) をコマンドプロンプト画面に出力するバッチファイル `kadai1-10.bat` を作成せよ。

課題 1-11. 工科大のホームページ (<http://www.niit.ac.jp/>) を開くバッチファイル `kadai1-11.bat` を作成せよ。

課題 1-12. 工科大のホームページと、この講義のホームページを開くバッチファイル `kadai1-12.bat` を作成せよ。なお、`%` はバッチファイル内では特別な意味を持つので、URL 文字列等で `%` をバッチファイル内で使う場合は、`%%` と書く必要がある。

課題 1-13. 次のことを順に実行するバッチファイル `init1.bat` を作成せよ。

1. コマンドプロンプトの背景色、文字色を設定する
2. コマンドプロンプト画面をクリアする
3. 計算機実習室の Web ページ (<http://www.niit.ac.jp/cc/>) を立ち上げる
4. `otbedit` を立ち上げる
5. 今日の日付と現在の時刻を表示する

(これを次回から実習の最初に実行すると良い。)

4 バッチファイルの変数

コマンドプロンプトでは、環境変数 と呼ばれる変数を使うことができる。環境変数はあらかじめ設定されているものもあり、それと同じ名前の変数を使ってしまうと、コンピュータの動作に影響をおよぼす場合があるので、定義済みの変数は使わないようにすること¹²。

¹²意味を十分知った上であれば、そうするのは構わない。

環境変数の表示、設定、削除には、set コマンドを使用する。

- set : 環境変数の一覧を表示
- set [変数名]=[値] : [変数名] の環境変数に [値] を設定
- set [変数名]= : [変数名] の環境変数を削除 (= の右に何も書かない)

C 言語とは違い変数の宣言は不要で、変数の値は基本的には文字列。文字列は引用符で囲む必要はない¹³。

バッチファイルはフリーフォーマットではないので、set コマンドの「=」の前後にスペースを入れてはいけない。スペースを入れると、変数名にスペースがついたり、値文字列にスペースが含まれたりするので、バグの原因となりがちである¹⁴。

環境変数の値は、コマンドプロンプトやバッチファイルでは「 %[環境変数名] %」で参照できる。例えば、

```
Z:> set a=竹の
Z:> set b=茂治
Z:> set c=%a% %b%
Z:> echo %c%
```

とすると変数 c の値は「竹の 茂治」となり、その結果が最後の echo コマンドで表示される。

課題 1-14. 環境変数の一覧を表示し、どのような環境変数があらかじめ設定されているか確認せよ。

課題 1-15. 変数 test1 に自分の名前を設定し、それを echo とその変数名を使って表示させるバッチファイル kadail-15.bat を作成し、実行せよ。

課題 1-16. 環境変数の一覧を表示し、変数 test1 が残っているか確認せよ。

課題 1-17. 変数 test1 を削除せよ。

課題 1-18. 変数 test1 に「新潟県」と設定し、test2 には「柏崎市」と設定し、その文字列をつなげた値を持つ変数 test3 を、変数名を使って作成せよ。

¹³逆に文字列を引用符で囲むと引用符自体もその文字列の一部となる。

¹⁴意味を知った上でやるのであれば構わない。

課題 1-19. 「set test1=abc」の後で「set test1=%test1%def」とするとどうなるか予想し、実際にそれをやって確認せよ。

5 バッチファイルのオプション

バッチファイルで使える特別な変数として、オプションを表す変数がある。バッチファイルにオプションを与えて

```
Z:> test1.bat 123 竹の 茂治
```

のように実行すると、このオプションの値はバッチファイル内では、指定した順に %1, %2, %3, ..., %9 のような名前で参照できる。%0 はバッチファイル名自身になり、%* は %1 以降のオプション文字列全体となる。例えば上の例では、

```
%1=123, %2=竹の, %3=茂治, %0=test1.bat,  
%*=123 竹の 茂治
```

となる。また、%4 以降はこの場合は空文字列となる。

課題 1-20. バッチファイルの 5 つまでのオプションパラメータを 1 行ずつ、指定した逆順に表示するバッチファイル kadai1-20.bat を作成せよ。

課題 1-21. 1 つ目と 2 つ目のオプション文字列を連結した文字列をコマンドプロンプト画面に出力するバッチファイル kadai1-21.bat を作成せよ。

コラム: MS-Windows のスクリプト環境

Unix には「小さいプログラムを使い回したり、組み合わせて使う」ための環境として、対話型シェル環境やシェルスクリプトがあるが、MS-Windows にもそれに近いものとして以下のものがある。

- コマンドプロンプト + バッチファイル (以下、バッチ環境)
- WSH (Windows Scripting Host)
- Windows PowerShell

	バッチ環境	WSH	PowerShell
対話環境	コマンドプロンプト	なし	ある (専用のシェルウィンドウ)
スクリプト言語	バッチファイル (.bat)	VBScript (.vbs), JScript (.js) 等	専用のスクリプト言語 (.ps1)
構文や変数	一応使える	使える	使える
外部コマンドやモジュール	一応ある	ある (オートメーションオブジェクト)	一応ある
パイプ、リダイレクション	ある (ファイルベース)	少なくとも簡単なものはなさそう	ある (オブジェクトベース)
実行ファイル	cmd.exe	wscript.exe (cscript.exe)	PowerShell.exe
XP では	そのまま使える	そのまま使える	要インストール
歴史	MS-DOS から	Windows 98 から	2007 年頃から

表 1: バッチ環境、WSH、PowerShell の特徴

これらの特徴や違いなどを表 1 に示す。

バッチ環境は、正確には MS-DOS の頃は command.com というインタプリタが使われていて、Windows NT 以降はその拡張版である cmd.exe に置きかわっている。現在は PowerShell への移行が推奨されているようである。

また、WSH も PowerShell への移行が推奨されていて、今後はメジャーバージョンアップもないという話であるが、PowerShell はまだ新しく、安定性や仕様の変更も含めて今後どうなるかはわからない。

よって、これらのうちでは、古くから使われていて、そして今後もなくならないと思われるバッチ環境を利用するのが一番ましな気がするが、以下の点などで Unix のシェルスクリプトにはかなり劣る。

- 「引用符」がちゃんと使える形にはなっていないので、文字列変数、コマンドのパラメータ等に関する制限がある。
- while 文、switch 文がないので if, goto に頼りがちになり、スパゲッティ化しやすい。
- フィルタやバッチファイル用に用意されているコマンドが少ない。
- ヒアドキュメントが書けない。
- 長いコマンドを複数行に分けて書けない。

よって、本当に実用的なものを作るには、バッチファイルと AWK やフリーソフトなどを組み合わせる必要があるだろう。

しかし簡単な処理なら C のようなプログラミング言語を使わなくてもバッチファイルで行えるし、既存のプログラムを組み合わせで自分用の便利な仕組みを簡単に作れること、ファイルとフォルダに関する操作もひとつひとつエクスプローラでやるより for 文などで簡単に行えることが多いこと、および C 言語で外部プログラムを利用する際もバッチファイルの知識が役に立つことなどから、バッチファイルやコマンドプロンプトのことを知ることは意味があると思う。