

rannu ユーザーズガイド

rannu コマンドは、乱数表を作成します。乱数は、二桁の十進数です。最小値、最大値を設定できます(たとえば、1 から 37 まで)。乱数の種(シード)を設定できます。



CSV

空白ではなくコンマを挿入して CSV を作成できます。

KB

キロバイト(KB)の単位で乱数表のサイズを指定できます。乱数表の行数を指定できます。

擬似乱数発生器

MT(メルセンヌツイスタ)および xorshift の擬似乱数発生器(PRNG)を実装しています。

6ビットずつ利用

MT から生成された乱数を 6 ビットずつ利用して、0 から 63 まで発生できます。

算術平均および分散

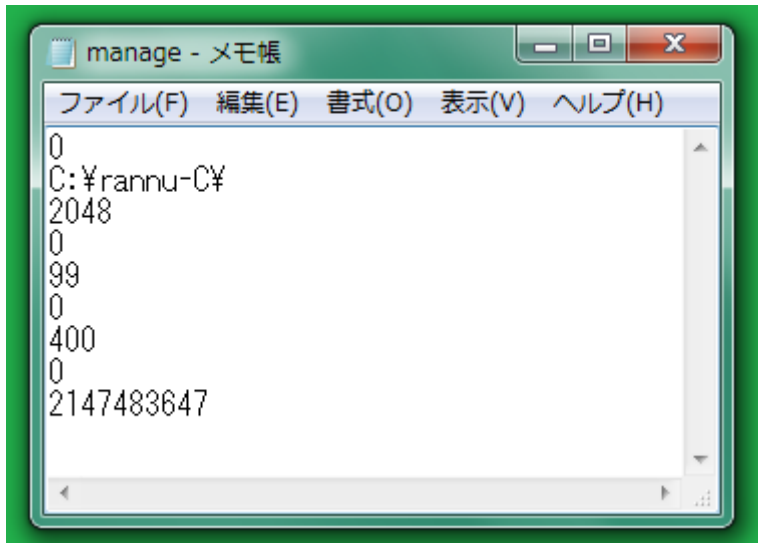
rannu コマンドは、二桁の数字をカウントして表を作成できます。算術平均および分散を計算できます。

32ビットアプリケーション

rannu コマンドは、Windows の 32 ビットアプリケーションです。

管理ファイル

コマンドプロンプトにコマンドおよびパラメータを入力して実行できます。rannu を設定するために、manage.txt という管理ファイルが用意されています。パラメータが無い場合、rannu コマンドは、manage.txt に従います。たとえば、manage.txt の第 1 行が 0 ではない場合、区切り文字としてコンマを使用します。すなわち、CSV になります。Windows アクセサリのメモ帳で manage.txt を表示した例(Windows 7)



6ビットずつ利用

MT から生成された乱数を 6 ビットずつ利用して、0 から 63 まで発生できます。

インストール

1. ロトサマリーまたはトリシーカー (<http://tori.tobihiro.jp/PDF-zip-7z.html>) というウェブサイトで rannu-C.zip をダウンロードしてください。
2. rannu-C.zip を展開してください。
3. 利用者は、フォルダの改名および移動が可能です。

プログラムの削除

アンインストール(プログラムの削除)を行うには、フォルダごと削除してください。

構築

Dev-C++の IDE で、C のプロジェクトとして、rannu を構築しました。

免責条項

rannu コマンドのご利用によって発生するいかなる損害も、Sogaya は、責任を負わないものとします。

著作権

作者として、Sogaya(そがや)は、rannu の著作権を保有しています。

ユーザーズガイド

2019 年 8 月 13 日。Revision 1.041

rannu コマンド

コマンドプロンプトに rannu コマンドおよびパラメータを入力して Enter を押してください。



省略

rannu コマンドを設定するために、manage.txt という管理ファイルが用意されています。パラメータが無い場合、rannu コマンドは、manage.txt に従います。

行番号	記入	パラメータ	説明
1	0		0 ではない場合、rannu.csv を作成します。
2	C:¥rannu-C¥		出力先のディレクトリ名、最後に¥が必要です。
3	32	-k32	32KB のファイル
4	1	-i1-37	最小値。一桁の場合、二桁になるように、十の位に0を付加します。
5	37		最大値
6	0		記入内容が0ではない場合、0から63まで、でたらめに書き出します。
7	400	-f400	400回のアイドリングを行う。
8	123456	-s123456	乱数の種(シード)
9	2147483647	-L-	ファイルの行数。
	20	-L20	

パラメータで編集されない

コマンドおよびパラメータで起動した場合、manage.txt にかかわらず、rannu コマンドは、パラメータに従います。manage.txt は、パラメータで編集されるものではありません。テキストエディタでmanage.txtを編集してください。さもないと、FLTK版のRannuを起動して、Manage ウィンドウで Save ボタンを押してください。

区間

最小値から最大値まで、でたらめに書き出すには、-i 最小値-最大値のパラメータを指定してください。たとえば、13 から 43 まで、でたらめに書き出す設定を行うには、コマンドプロンプトに下記のコマンドを入力して Enter を押してください。

rannu -i13-43

ビンゴ5およびロト

ビンゴ5、ミニロト、ロト6、ロト7の数字をでたらめに書き出すように設定するには、それぞれ -b, -m, -6, -7 を指定してください。たとえば、-b を指定した結果は、-i1-40 を指定した結果と同じになります。

111111b

0 から 63 まで、でたらめに書き出すように設定するには、rannu コマンドのパラメータとして -g を付加してください。この場合、rannu コマンドは、32 ビットを 6 回発生して、合計 192 ビットのうち、6 ビットずつ利用して書き出します。この処理を反復します。二進数として、111111b が最大値です。

$$63 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$$

6 ビットずつ利用

rannu コマンドは、Windows の 32 ビットアプリケーションですが、-g が指定された場合、rannu コマンドは、32 ビットではなく 6 ビットずつ利用して書き出します。-g を指定した結果は、下記のコマンドおよびパラメータと同じになります。

rannu -i0-63

シード

rannu コマンドは、MT および xorshift の擬似乱数発生器 (PRNG) を実装しています。PRNG のシードとして、1 から 999999 までの整数を想定しています。-s 数字を指定してください。

年月日時分秒

-s0 が指定された場合、rannu コマンドは、自動的に年月日時分秒に基づいてシードを提供します。rannu コマンドが提供したシードを画面に表示するには、さらに、-u を付加してください。下記の例では、まず、rannu コマンドにシードを提供させました。次に、そのシードを指定して、rannu コマンドを実行しました。同じシードの結果として、同じ乱数表が作成されました。

rannu -s0 -u

```
C:¥rannu-C>rannu -s0 -u
196611

C:¥rannu-C>COPY rannu.txt text.tmp /Y
      1 個のファイルをコピーしました。

C:¥rannu-C>rannu -s196611

C:¥rannu-C>FC text.tmp rannu.txt
ファイル text.tmp と RANNU.TXT を比較しています
FC: 相違点は検出されませんでした

C:¥rannu-C>
```

アイドリング

PRNG の設定として、シードだけではなくアイドリングも設定できます。たとえば、利用者がアイドリング回数を 400 に設定した場合、rannu コマンドは、シードに基づいて連続して発生した 400 個の乱数を無視します。-f400 が指定された場合、rannu コマンドは、401 回目以降を rannu.txt に書き出します。

算術平均

rannu.txt を読み込んで、二桁の数字をカウントして、表を作成して、算術平均 (arithmetic mean) を計算するには、rannu コマンドのパラメータとして、-e を付加してください。算術平均は、ファイルではなく画面に表示されます。

数字をカウントした表

-e が付加された場合、rannu コマンドは、各数字の個数の表を算術平均に追加します。表の追加を抑制するには、rannu コマンドのパラメータとして、さらに-t-を付加してください。

rannu -i0-99 -e -t-

```
C:¥rannu-C>rannu -i0-99 -e -t-  
49.487270
```

```
C:¥rannu-C>MORE rannu.txt
```

```
79 85 43 16 24 48 30 49 50 86 95 81 31 98 38 91 88 28 55 68  
25 91 05 33 31 35 01 03 34 41 89 13 46 64 16 35 82 81 86 50  
49 33 97 75 25 62 54 39 06 93 18 04 00 72 35 71 07 86 42 45  
80 31 83 25 89 07 16 24 65 37 29 95 12 54 23 23 29 31 67 60  
54 18 63 83 73 83 57 34 03 92 42 44 08 73 21 04 23 39 37 82  
23 59 08 99 76 40 16 05 07 45 57 02 72 31 00 16 65 08 38 13  
58 49 04 49 52 04 31 56 86 54 08 11 77 07 54 81 25 46 57 11  
86 60 81 79 11 06 08 58 16 01 46 26 43 41 17 02 39 71 72 52  
62 78 34 58 82 66 53 27 60 39 67 93 16 42 05 58 86 61 82 55  
14 99 70 69 53 24 78 19 97 75 34 72 49 92 34 47 05 73 95 31  
50 73 95 92 99 58 21 94 45 79 52 33 17 89 49 05 29 62 61 60  
82 38 25 65 07 12 35 90 19 02 62 08 24 03 64 01 54 27 95 95  
51 01 67 11 81 69 99 99 35 82 36 75 02 22 28 68 05 36 46 97  
49 58 79 01 90 30 27 83 75 87 19 74 85 71 54 61 24 75 82 54  
99 98 34 84 60 01 97 99 16 99 71 77 12 08 02 27 88 18 64 71  
97 02 40 10 83 39 95 93 47 73 54 16 10 56 95 88 86 09 45 95  
40 18 53 01 54 78 31 32 35 12 84 55 62 14 35 28 22 97 61 96  
09 30 78 11 52 70 49 13 01 36 75 66 89 30 00 24 99 05 82 06  
86 66 90 96 36 18 04 60 69 26 33 41 12 08 57 99 69 00 59 02  
29 14 57 35 37 21 57 88 78 62 75 25 42 30 47 57 75 94 13 02  
05 53 80 96 10 69 57 78 16 30 98 13 49 33 84 18 13 79 76 75  
80 70 35 37 42 24 30 66 86 54 84 19 07 52 15 27 93 30 02 74  
02 30 62 43 42 63 03 57 12 08 35 66 39 07 73 77 46 91 01 57  
14 46 26 24 66 77 79 96 47 89 33 78 44 95 33 10 26 65 71 38  
-- More (0%) --
```

分散

rannu.txt を読み込んで、二桁の数字をカウントして、表を作成して、分散 (variance) を計算するには、rannu コマンドのパラメータとして、-a を付加してください。分散は、ファイルではなく画面に表示されます。

数字をカウントした表

-a が付加された場合、rannu コマンドは、各数字の個数の表を分散に追加します。表の追加を抑制するには、rannu コマンドのパラメータとして、さらに-t を付加してください。

標準偏差

標準偏差は、分散の平方根です。

期待値

平均を計算してから、分散を計算するのではなく、期待値を利用して分散を計算しま

す。たとえば、81 から 90 まで、でたらめに書き出すならば、期待値は、85.5 です。

$$\text{総和} = (\text{最小値} + \text{最大値}) \times \text{個数} / 2$$

$$\text{確率} = 1 / \text{個数}$$

$$\text{期待値} = (\text{最小値} + \text{最大値}) / 2$$

xorshift

利用者が PRNG の実装を指定しなかった場合、rannu コマンドは、自動的に MT を選択します。PRNG の実装として xorshift を選択するには、-px を指定してください。下記の例では、まず、MT による結果をカウントして、算術平均を表示しました。次に、xorshift による結果をカウントして、算術平均を表示しました。さらに、xorshift による結果をカウントして、分散を表示しました。最後に、MT による結果をカウントして、分散を表示しました。シードは同一ですが、MT と xorshift は、結果が相違しました。

```
C:\¥rannu-C>TYPE manage.txt
0
C:\¥rannu-C¥
2048
0
99
0
400
0
2147483647

C:\¥rannu-C>rannu -s567890 -e -t-
49.475492

C:\¥rannu-C>rannu -s567890 -e -t- -px
49.476377

C:\¥rannu-C>rannu -s567890 -a -t- -px
834.125917

C:\¥rannu-C>rannu -s567890 -a -t-
832.710886

C:\¥rannu-C>_
```

パラメータ	ほぼ同等な入力	説明
-6	-i1-43	ロト6
-7	-i1-37	ロト7
-a		分散および表を画面に表示する。
-a -t-		分散のみ表示する。
-b	-i1-40	ビンゴ5
-e		算術平均および表を画面に表示する。
-e -t-		算術平均のみ表示する。
-f400		400 回のアイドリングを行う。
-g	-i0-63	0 から 63 まで、でたらめに書き出す。
-i0-99		0 から 99 まで、でたらめに書き出す。
-k32		32KB(キロバイト)の乱数表を作成する。
-L20		20 行の乱数表を作成する。
-L-	-L2147483647	行数ではなくキロバイトでサイズを指定する。
-m	-i1-31	ミニロト
-pm		MT
-px		xorshift
-s0		rannu コマンドが自動的にシードを提供する。
-s0 -u		rannu 提供のシードを表示する。
-s123456		利用者がシードを提供する。
-V		バージョン番号を表示する。
-VC		短い説明を表示する。

キロバイト

rannu コマンドは、`rannu.txt` のサイズが KB(キロバイト)の単位で指定されることを想定しています。32KB の `rannu.txt` を作成するには、`-k32` を指定してください。

行数を無視したい

行数ではなくキロバイトでファイルのサイズを指定したいことを明示するには、`-L-` を指定してください。`-L-` を指定した結果は、行数として 2147483647 を指定した結果と同じになります。`rannu.txt` の行数がこうした巨大な数に達する前に、rannu コマンドは、指定のキロバイトで書き出しを終了します。

行数

`-k1` が指定された場合、rannu コマンドは、17 行を書き出します。ただし、第 17 行は、20 個ではなく 11 個になります。少ない行数を書き出すように設定するには、`-L` 数字を

指定してください。

遅いと感じるおそれ

rannu.txt のサイズが指定のキロバイトに達したとき、rannu コマンドは、乱数の書き出しを終了します。サイズを調節するパラメータとして、行数は、実行速度がキロバイトより遅い。大きい乱数表を書き出したい場合、行数ではなくキロバイトで指定してください。-Lを指定した結果は、-L2147483647 を指定した結果と同じになります。

キロバイト	パラメータ	rannu.txt の行数
1	-k1	17
16	-k16	265
128	-k128	2115
1024	-k1024	16913
10240	-k10240	169126

ファイルの行数

添付の atnd コマンドは、ファイルの行数をカウントできます。ファイルの行数を表示するには、atnd コマンドのパラメータとして、-LF を付加してください。ファイル名および行数が表示されます。また、最長の行の長さが記録されており、バイト数および行番号が表示されます。さらに、第 1 行が表示されます。第 1 行ではなく第 9 行を表示するには、atnd コマンドのパラメータとして、-p9 を付加してください。

```
C:¥Q>atnd -LF -p9 nineTen.txt
nineTen.txt: 17 lines
maximum LF-to-LF span of 30 bytes (line-9)
123456789,1234567890,123456789
C:¥Q>_
```

rannu.txt

まず、rannu コマンドに-k キロバイト数を付加しました。次に、atnd コマンドで、rannu.txt の行数をカウントしてみました。

```
C:¥rannu-C>rannu -k1

C:¥rannu-C>atnd -LF rannu.txt
rannu.txt: 17 lines
maximum LF-to-LF span of 60 bytes (line-1)
04 94 94 31 24 44 95 58 12 97 81 03 95 31 83 17 82 54 90 20

C:¥rannu-C>rannu -k16

C:¥rannu-C>atnd -LF rannu.txt
rannu.txt: 265 lines
maximum LF-to-LF span of 60 bytes (line-1)
66 48 94 66 99 57 56 76 82 76 20 35 46 46 18 12 32 39 81 61

C:¥rannu-C>rannu -k128

C:¥rannu-C>atnd -LF rannu.txt
rannu.txt: 2115 lines
maximum LF-to-LF span of 60 bytes (line-1)
72 50 91 08 41 51 09 44 64 89 89 47 47 52 85 27 67 26 82 16

C:¥rannu-C>rannu -k1024

C:¥rannu-C>atnd -LF rannu.txt
rannu.txt: 16913 lines
maximum LF-to-LF span of 60 bytes (line-1)
75 35 54 12 08 80 74 19 89 54 59 03 54 47 04 11 03 36 84 59

C:¥rannu-C>rannu -k10240

C:¥rannu-C>atnd -LF rannu.txt
rannu.txt: 169126 lines
maximum LF-to-LF span of 60 bytes (line-1)
62 01 98 64 62 52 53 45 97 45 51 28 43 84 78 28 10 39 75 02
```

UNIX の tail コマンドに似ている

ファイルの終了部分にある 5 行を表示するには、atnd コマンドのパラメータとして、-t5 を付加してください。atnd コマンドは、自動的にファイルの行数をカウントします。たとえば、1024 行である場合、-t5 は、-b4 -p1024 を意味します。atnd -t は、UNIX の tail コマンドに似ています。

atnd -t5 LineNo.txt

```
C:¥Q>atnd -LF LineNo.txt
LineNo.txt: 1024 line
maximum LF-to-LF span of 34 bytes (line-1000)
Line-1: 639 lines to Line-640

C:¥Q>atnd -b4 -p1024 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>atnd -t5 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>
```

UNIX の head コマンドに似ている

ファイルの開始部分にある 9 行を表示するには、atnd コマンドのパラメータとして、-h9 を付加してください。-h9 は、-b8 -p9 を意味します。atnd -h は、UNIX の head コマンドに似ています。

atnd -h9 manage.txt

行番号

行の先頭に行番号を表示するには、atnd コマンドのパラメータとして、-n を付加してください。正確に言えば、ファイルの行の番号ではなく、atnd コマンドが読み込んだ行の通し番号が表示されます。

atnd -n -h9 manage.txt

```
C:¥rannu-C>atnd -n -h9 manage.txt
1|0
2|C:¥rannu-C¥
3|2048
4|0
5|99
6|0
7|400
8|0
9|2147483647
C:¥rannu-C>
```

多数のファイルを連結する

-n が付加された atnd コマンドは、ファイルの行の番号ではなく、atnd コマンドが読み込んだ行の通し番号を表示します。たとえば、2 個のファイルを連結して読み込んだら、いわゆる行番号がずれます。UNIX の cat コマンドは、GnuWin32 のコアなユーティリティー (coreutils) に含まれています。

プログラミング言語 awk

インターネットで awk を検索してダウンロードできます。AWK, gawk, mawk, nawk など、どの awk も無料です。たとえば、GnuWin32 には、nawk があります。nawk をダウンロードして、インストールして、nawk.exe を awk.exe に改名することで、awk コマンドを実行できます。awk は、テキストファイルの行を入力して、行がパターンに合致した場合、行を処理します。利用者は、awk のスクリプトにどう処理するか記述します。スクリプトとは、プログラムのことです。

cat コマンドと同等な処理

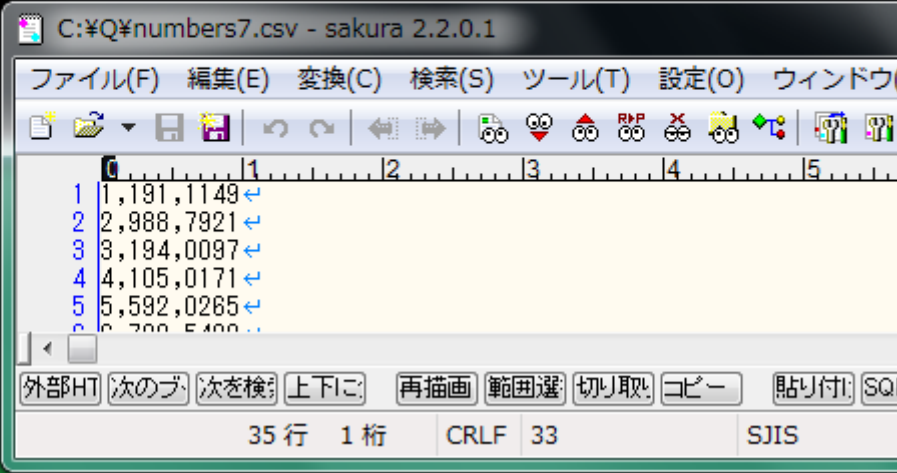
下記の例では、非常に短いプログラムであるため、スクリプトのファイルを用意しないで、awk コマンドのパラメータとして付加しました。cat コマンドと同様に、2 個のファイルを連結します。awk コマンドおよびパラメータの例

```
awk "{print}" m.txt n.csv
```

```
C:¥Q>cat marunana.txt numbers7.csv | atnd -n -h9
1|ナ
2|ナ
3|菜
4|㊦
5|Ⅶ
6|1,191,1149
7|2,988,7921
8|3,194,0097
9|4,105,0171

C:¥Q>awk "{print}" marunana.txt numbers7.csv | atnd -n -h9
1|ナ
2|ナ
3|菜
4|㊦
5|Ⅶ
6|1,191,1149
7|2,988,7921
8|3,194,0097
9|4,105,0171

C:¥Q>
```



Line	Content
1	1,191,1149
2	2,988,7921
3	3,194,0097
4	4,105,0171
5	5,592,0265

TYPE コマンドで連結する例

どの Windows も最初から TYPE コマンドを用意しています。下記の例では、標準エラー出力から NUL にリダイレクトすることでファイル名を消去します。

```
TYPE marunana.txt numbers7.csv 2> NUL | atnd -n -h9
```

短い説明

atnd コマンドに関する短い説明を表示するには、atnd コマンドのパラメータとして、-VC を付加してください。

```

C:\rannu-C>atnd -VC
atnd C-1.082

atnd -a2 -p567 file      # 2 lines after Line-567
atnd -b3 -p567 file      # 3 lines before Line-567
atnd -c4 -p567 file      # 4 lines before Line-567 and 4 lines after Line-567
atnd -c4 -p567 -n file   # A number appears at the beginning of each line.
atnd -h8 file            # Line-1, Line-2, Line-3, ..., Line-8
atnd -h8 < file          # Line-1, Line-2, Line-3, ..., Line-8
atnd -i90-123 file       # Line-90, Line-91, Line-92, ..., Line-123
atnd -LF -p567 file      # Count lines. Record the max span from LF to LF.
atnd -LF -p567 -s, file  # Record the max span from separator to separator.
atnd -LF -p567 -s¥x2C file # Record the max span from separator to separator.
atnd -p567 file          # Line-567
atnd -p567 -l file       # Show how large the buffer is.
atnd -t9 file            # 9 lines before EOF
atnd -V                  # version
atnd -VC                 # version with comments
atnd -z80                # Use a buffer of 80 bytes.
DIR | atnd -LF -=DIR -p9 # Display a label (DIR) instead of files.

C:\rannu-C>

```

パラメータ	ほぼ同等な入力	説明
-a2		後続の 2 行を表示する。
-b3		先行する 3 行を表示する。
-c4		先行する 4 行および後続の 4 行を表示する。
-f		どの行も同じ欄数を有しているか点検する。
-h8	-b7 -p8	第 1 行から第 8 行まで表示する。
-i90-123	-b33 -p123	第 90 行から第 123 行まで表示する。
	-p90 -a33	
-LF		行数をカウントする。最長の行のバイト数を記録する。
-LF -=DIR		ラベルとして DIR を表示する。
-LF -s,		コンマどうしの間にある欄の長さの最大値を記録する。
-LF -s¥x7C		2 桁の十六進数で区切り文字を指定する。
-n		行の先頭に番号を表示する。
-p567		第 567 行を表示する。
-t5	-b4 -p 行数	ファイルの終わりにある 5 行を表示する。
-V		バージョン番号を表示する。
-VC		バージョン番号、短い説明を表示する。

標準入力から読み込む

ファイル名が指定されていない場合、atnd コマンドは、標準入力(stdin)から読み込みます。利用者は、情報源(標準入力)をラベルとして命名できます。下記の例では、DIR というラベルを入力しました。

ファイルにコマンドの出力をリダイレクト

下記の例では、まず、DIR コマンドの出力を画面から一時ファイル(text.tmp)に、リダイレクトしました。次に、ファイルの行数をカウントするために-LF を付加して atnd コマンドを実行しました。ファイル名として、text.tmp が表示されました。

ファイル名ではなく stdin を表示

一時ファイルに、リダイレクトするのではなく、DIR コマンドの出力を標準入力として利用できます。すなわち、DIR コマンドの出力を atnd コマンドに入力するパイプを構成しました。情報源として、ファイル名ではなく stdin が表示されました。

stdin ではなく DIR を表示

最後に、stdin ではなく DIR が表示されるように、-=DIR を付加した atnd コマンドによるパイプを構成しました。

```
C:¥sjisEUCjpCS>DIR *.txt > text.tmp
C:¥sjisEUCjpCS>atnd -LF -p11 text.tmp
text.tmp: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥sjisEUCjpCS>DIR *.txt | atnd -LF -p11
stdin: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥sjisEUCjpCS>DIR *.txt | atnd -LF -p11 -=DIR
DIR: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥sjisEUCjpCS>_
```