文字列探索プログラム

著者：関 勝寿　公開日：2024 年 8 月 25 日　ソース：https://sekika.github.io/2024/08/25/findstring/

ディレクトリ内のファイルに特定の文字列を検索するプログラム findstring を作成した。grep -rI コマンドに似た動作をするが、PDF および DOCX ファイルの読み取りと検索機能を追加した。

1. インストール

```
pip install findstring
```

2. 使用方法

```
findstring [ オプション ] 検索文字列
```

3. 引数

・検索文字列：ファイル内で検索する文字列。

3.1. 省略可能な引数

・-h, --help: ヘルプメッセージを表示し、終了します。このオプションは、利用可能なすべてのオプションの説明を含む findstring の使用方法の概要を提供します。
・-b, --binary: バイナリファイルもスキャンします。このオプションを使用すると、ツールはバイナリファイルを読み取り、指定された文字列を検索しようとします。
・-d, --directory: 検索を開始するルートディレクトリ。指定しない場合、デフォルトで現在のディレクトリ (.) が使用されます。
・-t, --text: 出力に検索文字列を含む一致行を表示します。
・-l, --max_length: 結果として表示される最大文字数。デフォルトは 0 で、制限がないことを意味します。--max_length が指定されると、--text も有効になります。
・-v, --verbose: 詳細出力を有効にします。プログラムは、どのディレクトリやファイルがチェックされているかなど、動作に関するより詳細な情報を提供します。

4. 機能

・**PDF サポート**： プログラムは、pdfminer ライブラリを使用して PDF ファイル内を検索できます。PDF からテキストを抽出し、指定された文字列を検索します。
・**DOCX サポート**： DOCX ファイルもサポートしており、テキスト抽出は docx ライブラリで処理されます。
・**バイナリファイルのスキャン**： --binary フラグが有効な場合、プログラムはバイナリファイルを読み取り、指定された文字列を検索しようとします。
・**コンテキスト表示**： --text オプションが有効な場合、ツールは検索文字列を含む行を、--max_length オプションで指定された最大文字数で表示します。

5. 使用例

5.1. 現在のディレクトリで文字列を検索する

```
findstring "example_string"
```

5.2. 特定のディレクトリで文字列を検索する

```
findstring -d /path/to/directory "example_string"
```

5.3. 詳細出力を有効にして一致する行を表示する

```
findstring -tv "example_string"
```

5.4. 出力テキストの長さを 50 文字に制限する

```
findstring -l 50 "example_string"
```

5.5. バイナリファイルで検索する

```
findstring -b "example_string"
```

6. エラーハンドリング

プログラムは、読み取れないファイルなどのエラーを適切に処理するように努めます。ファイルの読み取り中にエラーが発生した場合、プログラムはそのファイルをスキップし、残りの処理を続行します。詳細モードが有効な場合は、エラーメッセージが表示されることがあります。

7. 検索結果のハイライト

プログラムは、デフォルトで一致する検索結果を赤色の太字で強調表示するように設定されています。このハイライトは、GREP_COLORS 環境変数、特に mt= オプションによって制御されます。

ハイライトされたテキストの色やスタイルを変更したい場合は、GREP_COLORS 環境変数で mt= 設定を変更できます。mt= 値は、マッチングテキストに使用されるスタイルと色を指定するカラーコードです。

例：

・**赤の太字 (デフォルト)**: mt=1;31
・**緑の太字**：mt=1;32
・**青の下線**：mt=4;34

カスタムカラーを適用するには、次のようにシェルで GREP_COLORS 環境変数を設定します：

```
export GREP_COLORS='mt=1;32'
```

この例では、ハイライトされたテキストが緑の太字に変更されます。

プログラムは出力が端末 (TTY) に向けられているかどうかを自動的に検出します。そうでない場合、色付けなしでテキストを表示します。

8. ライブラリとして使用する

findstring をライブラリとして使用することもできます。

```
from findstring import findstring
findstring("/path/to/directory" "example_string")
```

findstring 関数の引数は以下のとおりです。

・**root_dir** (str): 検索を開始するルートディレクトリ。
・**search_string** (str): ファイル内で検索する文字列。
・**verbose** (bool, 任意): True に設定すると、検索プロセス中に追加情報が表示されます。デフォルトは False です。
・**show_text** (bool, 任意): True に設定すると、検索文字列を含む一致する行が表示されます。デフォルトは False です。
・**max_length** (int, 任意): 結果に表示する文字数の最大値。デフォルトは 0（制限なし）です。
・**binary** (bool, 任意): True に設定すると、バイナリファイルもス

キャンされます。デフォルトは `False` です。