

Python による C 言語デバッグ支援ツールの開発

日大生産工 (学部) 〇石山 綾乃
日大生産工 新井 雅之

1 まえがき

ソフトウェアの開発工程の半分以上はデバッグ作業によって占められるといわれており、デバッグ作業の効率化は重要な課題である。特に、プログラミング教育の重要性が近年高くなっているため、初学者向け支援システムの早急な開発が求められている。情報を計測して支援することによるデバッグ作業効率の向上を目的とした研究は過去にも多くあるが[1]、これらの研究は開発現場での支援が目的のことが多く、初学者向けの教育支援目的は少ない[2]。

松村は、C 言語初学者の支援を目的として、バグレポート採取およびデバッグ支援ツールを C 言語を用いて開発した[3]。本研究では python を用いて、C 言語初学者を対象としたバグレポート採取並びにデバッグ支援ツールを開発する。python を用いることにより、先行研究と比較して文字列の処理が簡素化され、開発などが容易になると考えられる。システム全体の設計を行い、一部の機能を評価システムとして実装し、動作を確認した。

2 デバッグ支援ツールの設計

2.1 デバッグ支援ツールの概要

本デバッグ支援ツールは、指定されたソースファイルを読み込み、検査項目毎に順にバグ検出を行う。システムの動作の概略を図 1 に示す。バグを検出した場合は、検出された行に対して、検出位置及び検出内容を出力ウィンドウに表示する。また、バグレポートファイルを作成し、ファイル名、日時、異常があった内容、件数を出力する。

現在の実装では本システムはコマンドラインツールとして動作する。今後、先行研究[2]と同様にデバッグ支援ツールを Visual Studio のプリプロセッサとして動作するよう設定し、ビルドイベント機能から実行させる予定である。

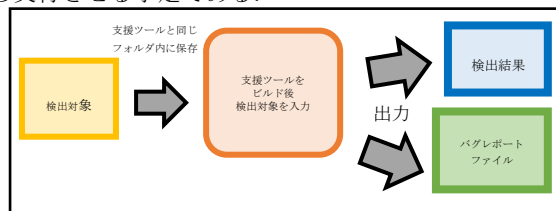


図 1. デバッグ支援ツール図説

2.2 検査項目

デバッグ支援ツールの検査項目として、初学者が見落としやすい以下の点を対象とした。

(1) scanf におけるアンパサンド ‘&’ 検出

この検査項目では、scanf の記述を検索し、引数が“&変数名”となっていない場合、また適切な構文になっていない場合記述ミスと判断し警告を出力する。ただし変数が文字列の場合は、文字列配列であり ‘&’ が必要ないため警告出力対象には含めない。図 2 に検出対象となる記述の例を示す。図 2 では C 言語において“%df”という記述は存在しないためバグの要因となる。

(2) 全角文字及び空白の検出

この検査項目は不適切に使用されている全角文字ならびに全角の空白を検出し、全角文字の混入位置を出力する。図 2 では 7 行目に全角の空白、9 行目に全角文字が“0”の範囲外に存在するためバグの要因となる。

```

#include<stdio. h>
int main(void)
{
    int c;
    printf("1 つ目を入力してください");
    scanf("%d", &c);
    printf("2 つ目を入力して下さい");
    scanf("%df", &a);
    printf("hello. ");
    printf("hello, world¥n");
    printf("おはよう . . . ¥n");
    printf("おはようございます");
    return 0;
}
  
```

図 2. 検査項目 (1), (2) の誤りがあるソースコードの例

(3) 変数宣言及び初期化検出

この検査項目では未宣言、または初期化されていない変数の使用の検出を行う。C 言語においては宣言されていない変数を用いるとバグが出てしまう。宣言されていない場合、警告を出力する。

(4) if 文の条件式記述不備検出

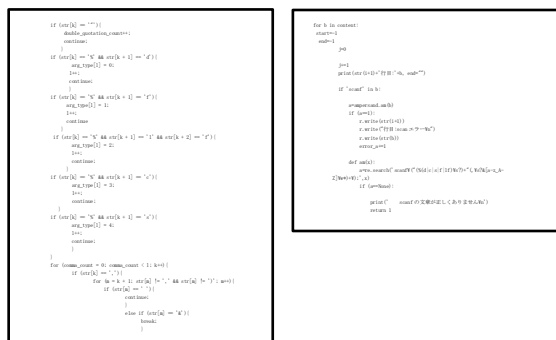
“変数 a, b の値が等しいならば,”と記述する際、if (a = b)と記述するのは誤りである。本検査では if 文の記述を検索し“==”のスペルミスだと判断した際、警告を出力する。

(5) for 文の条件式記述不備検出

この検査項目では、for 文の条件式がセミコロン ‘;’ で区切られているかを検査する。コロンやカンマによる区切りを検出した場合、また区切りがされていない場合、警告を出力する。

3 実装と動作検証

本研究では Visual Studio を開発環境として用い、python アプリケーションとして実装した。現在、検証項目の(1) scanf におけるアンパサンド ‘&’ 検出、(2) 全角文字及び空白の検出を出力する機能と、バグレポートファイルを出力する機能を搭載している。



(a) C 言語

(b) python

図 3. Scanf 部分の検出ソースコード

図 3 に検出項目(1)scanf のソースコードの一部を示す。図 3 (a)は C 言語を用いた先行研究のソースコードの一部、図 3 (b)は python を用いた本研究のソースコードの一部である。図 3 (a)では条件式を多く用いているが、図 3 (b)では正規構文を用いることによって 1 個の条件式で検出が可能となり記述が簡略化された。

図 2 のプログラムを対象としてファイル名を入力しデバッグ支援ツールを行った結果を図 4 に示す。本研究では松村が行っていたプリプロセッサとして動作するようにする設定を行っておらず、本支援ツール内のフォルダ内に検出対象のファイルを移動させる必要がある。その後実行し、出力ウィンドウに表示された検出結果を図 4 に示す。図 4 より、文章が 1 行ずつ出力されている中で scanf の中にバグとなる“%df”と書かれた 8 行目、全角空白が含まれている 10 行目と全角文字“を”が含まれている 12 行目のバグが検出されていることを確認した。

出力と同時に作成されたバグレポートファイルの記述を図 5 に示す。図 4 のバグ検出と同様に各項目の検出箇所、バグ内容、加えてバグ数と日時が出力され、8 行目と 10 行目、12 行目でバグが検出されてい

ることが確認できた。また、バグレポートファイルは出力されている日時と同様のファイル名<bug_report20191014_114433.txt>となっていることが確認できた。

```
1 行目:#include<stdio. h>
2 行目:int main(void)
3 行目:{
4 行目:int c;
5 行目:printf(“1 つ目を入力してください”);
6 行目:scanf(“%d”, &c);
7 行目:printf(“2 つ目を入力して下さい”);
8 行目:scanf(“%df”, &a);
    scanf の文章が正しくありません
9 行目:printf(“hello. ”);
10 行目:printf(“hello, world\n”);
    :全角が含まれています
11 行目:printf(“おはよう . . . \n”);
12 行目:printf(“おはようございます”を);
    を:全角が含まれています
13 行目:return 0;
14 行目;}
```

図 4. デバッグ支援ツールの出力結果

20191014 114433

Test. c

8 行目:scan バグ

scanf(“%df”, &a);

10 行目:全角バグ

printf(“hello, world\n”);

12 行目:全角バグ

printf(“おはようございます”を);

全角バグ : 2

Scanf バグ : 1

図 5. バグレポートファイルの出力

4 まとめ、今後の課題

本研究では、Python を用いた C 言語初学者デバッグ支援ツールを開発しバグの出力及びバグレポートファイルの動作確認を行った。今後はプリプロセッサとして動作させるよう設定をしていく。また、検出項目を増やす予定である。

参考文献

- [1] 瀧本恵介 他, “原型分析を用いたソフトウェアバグ分析,” 電子情報通信学会技術研究報告, No. DC2016-35, pp. 91-96, 2016 年.
- [2] 江本鶴子, 竹内章 “プログラミング初心者にトレースを指導するデバッグ支援システムの開発と研究,” 日本教育工学会論文誌, Vol.32, No. 4, pp.369-381, 2009 年.
- [3] 松村恵汰, “バグレポート採取のための C 言語初学者デバッグ支援ツールの開発,” 日本大学生産工学部数理情報工学科提出卒業論文, 2019 年.