

テスト報告書

1. ブラックボックステストのテスト項目

・エラー.txt

(1) テスト項目の説明

引数に指定されているファイル名に日本語が含まれている場合はエラーメッセージを表示してプログラムを終了するという仕様に基づき、日本語を含むファイルを引数に指定された場合にエラーメッセージを表示するかを調べるためのテストである。

(2) テストケースと予想される結果の説明

ファイル名に日本語を含むエラー.txt を実行すると、
error:filename is japanese
という表示が標準出力に出力されプログラムを終了すると予想される。

(3) テスト実行の結果

エラーメッセージを出力せずにファイルの出力を行ってしまった。

(4) 対処

メインモジュール内でエラー処理に関する記述を追加した。修正後、同じテスト実行を行うと、
error:filename is japanese
というエラーメッセージが標準出力に出力されたため予想される結果と同様の結果となった。

・preserve.c

(1) テスト項目の説明

C プログラムの場合予約語をクーリエボールド(Courier-Bold, 英文用等幅太字フォント)で表示するという仕様に基づき、C ファイルにおけるすべての予約語がクーリエボールドで表示されるかを調べるためのテストである。また、コメント中の予約語はクーリエボールドで表示しないという仕様のためコメント内の予約語がクーリエボールドにならないことも同時に調べる。

(2) テストケースと予想される結果の説明

C ファイル **preserve.c** 内には C プログラムのすべての予約語が記述してある。
またコメント内の予約語の出力をテストするために、

1. (すべての予約語)
2. '(すべての予約語)'
3. “(すべての予約語)”
4. //(すべての予約語)
5. /*(すべての予約語)*/

の5パターンを準備した。

予想される結果としては1.のみがクーリエボールドで表示され、それ以外はすべてクーリエ(Courier, 英文用等幅フォント)で出力される。

(3) テスト実行の結果

1.のみがクーリエボールドで出力されそれ以外はすべてクーリエで出力されたため予想される結果と同様の結果となった。

(4) 対処

特になし

・signs.c, signs.txt

(1) テスト項目の説明

すべての記号および数字が正確に出力されるかのテストである。

(2) テストケースと予想される結果の説明

C プログラム signs.c およびテキストファイル signs.txt を用意し、それぞれのファイルにすべての記号および数字を記述する。

予想される結果としては両ファイルともにすべての記号および数字が正しく出力される。

(3) テスト実行の結果

両ファイルとも“?”の記号のみうまく出力されていなかった。

(4) 対処

LEX モジュールにおいて“?”の記号の処理が抜けていたためその処理を追加することですべての記号および数字が両ファイルで正しく出力されるようになった。

・58.txt

(1) テスト項目の説明

1 ページの最大行数が 58 行という仕様に基つき、58 行ちょうどのファイルをうけとったときに無駄な改ページが出力されないことを調べるテストである。

(2) テストケースと予想される結果の説明

58 行のファイル 58.txt を用意する。

予想される結果としては改ページが行われずに 1 ページ 58 行のファイルが出力される。

(3) テスト実行の結果

1 ページ 58 行のファイルが出力され改ページが行われなかったため予想される結果と同様の結果となった。

(4) 対処

特になし

・1moji.c.txt

(1) テスト項目の説明

1 文字のみが入力されたファイルを受け取ったときに正しく出力を行うかをテストする。
また、拡張子の指定を正確に受け取れているかをテストする。

(2) テストケースと予想される結果の説明

改行文字 1 文字のみを入力したテキストファイル 1moji.c.txt を用意する。また、ファイル名を 1moji.c.txt のように拡張子の判別がしにくい形にしたファイル名にした。

予想される結果としては、ファイル名をテキストファイルとして受け取り、改行を 1 回分出力する。

(3) テスト実行の結果

ファイル名をテキストファイルとして受け取り改行を 1 回出力したため予想される結果と同様の結果となった。

(4) 対処

特になし

・longlonglonglonglonglong.tt

(1)テスト項目の説明

1 行の文字数が80文字であるかどうか,1 ページが58行であるかどうか,改ページをした際に行番号の繰り上がりが対応しているかどうか,1 万ページをこえるとページ番号が 00000,00001...となっているかどうか,行番号が 9999 をこえると 0000,0001...となっているかどうかを調べる.

(2)テストケースと予想される結果の説明

テストケースとして、A~Z までの文字列を 10000×1000 用意した.予想される結果は 1 行の文字数が80 文字,1 ページが58行,改ページされると行番号が繰り上がり,99999 ページをこえると次は 00000 から始まり,行番号が 9999 行をこえると 0000 から始まると予想される.

(3)テスト実行の結果

1 行の文字数が80文字,1 ページが58行,改ページされると行番号が繰り上がり,99999 ページをこえると次は 00000 から始まり,行番号が 9999 行をこえると 0000 から始まった.

(4) 対処

特になし

・engjap.txt

(1)テスト項目の説明

全角文字が 2 文字でカウントされているかどうかを調べる.また80文字目に全角文字がきた場合に改行されるかどうかを調べる.

(2)テストケースと予想される結果の説明

テストケースは全角文字 45 文字の文字列と,全角文字で始まり,全角と半角を交互に交えた 90 文字の文字列と半角文字角で始まり,全角と半角を交互に交えた 90 文字の文字列を用意した.予想される結果は全角文字 45 文字の文字列では,40 文字目で改行され,全角文字で始まり,全角と半角を交えた90文字の文字列では53 文字目で改行され,半角文字で始まり,全角と半角を交えた90文字の文字列でも53文字目で改行させると予想される.

(3)テスト実行の結果

全角文字 45 文字の文字列では 40 文字目で改行され,全角文字で始まり,全角と半角を交互に交えた 90 文字の文字列では 53 文字目で改行され,半角文字角で始まり,全角と半角を交互に交えた 90 文字の文字列では53文字目で改行された.

(4) 対処

特になし

・tabtest.h , tabtest.txt

(1)テスト項目の説明

タブ文字を入力した際に,8 の倍数文字目までスペースが埋められているかどうかを調べる.

(2)テストケースと予想される結果の説明

テストケースには、タブの後に次の文字がくる場合、一文字後にタブがきてその後文字がくる場合、二文字後にタブがきてその後文字がくる場合、三文字後にタブがきてその後文字がくる場合、四文字後にタブがきてその後文字がくる場合、五文字後にタブがきてその後文字がくる場合、六文字後にタブがきてその後文字がくる場合、七文字後にタブがきてその後文字がくる場合、八文字後にタブがきてその後文字がくる場合の九つの場合を考えてテストを行う.

タブが入力されていると,8 の倍数文字目までスペースで埋められるという結果が予想される.

(3)テスト実行の結果

タブが入力されていると8の倍数文字目までスペースが埋められていた。

(4) 対処
特になし

2. ホワイトボックステストのテスト項目

[illegible]

(1) テスト項目の説明

main モジュールでテキストファイルに分岐し,header モジュールでファイル名が長い場合の分岐,linemanage モジュールでは行数が 10000 行をこえる分岐以外の分岐,他のモジュールでは全分岐網羅するようなファイルのテストを行う.

(2) テストケースと予想される結果の説明

[illegible]

(3) テスト実行の結果

バグが起こることなく出力が得られた。

(4)対処
特になし

[illegible]

(1) テスト項目の説明

main モジュールで c ファイルに分岐し,header モジュールでファイル名が長い場合の分岐,linemanage モジュールでは行数が 10000 行をこえる分岐以外の分岐,他のモジュールでは全分岐網羅するようなファイルのテストを行う.

(2) テストケースと予想される結果の説明

[illegible]

(3) テスト実行の結果

バグが起こることなく出力が得られた。

(4)対処
特になし

- **white.c**

(1) テスト項目の説明

main モジュールで c ファイルに分岐し, header モジュールでファイル名が短い場合の分岐, linemanage モジュールでは行数が 10000 行をこえる分岐以外の分岐, 他のモジュールでは全分岐網羅するようなファイルのテストを行う.

(2) テストケースと予想される結果の説明

(1)の仕様をすべて満たすファイル **white.c** を作成した。
 予想される結果としてはバグが起こることなく出力が完了することである。

(3) テスト実行の結果

バグが起こることなく出力が得られた。

(4) 対処

特になし

3. テストフェーズ全体で発見できた誤りの総数

テストフェーズ全体で 2 つの誤りを発見できた。