end-of-fileについて

情報科学演習C ミニレクチャ

内山 彰

end-of-file

- ファイルの終端を表す。実態はシステムで定義されている整数値(int)で、char型に現れない値。

- -1であることが多いが処理系によって値が異なるので、EOFマクロを使って判定すること。

- char: 1 byte

- int: 2byte以上（演習室では4byte）

- charにEOFを代入してはいけない！（そもそも型が違う）

charにEOFを代入すると・・・

- EOFが-1(0xff)の場合、chには0xffが代入される
- EOFとの比較ではcharをintに変換してから比較する
  - 多くの環境では0xffをintに変換しても-1なのでうまくいく
- 問題が発生する場合もある
  - 符号無しcharの場合に-1が表現できない
    - 符号付きchar：intにすると-128〜127
    - 符号無しchar：intにすると0〜255
  - 0xffという文字が読み込まれるとEOFと判断
    - 0xff(1byte)と0xffffffff(4byte)は違う！
- getcharなどはintを返すので、右図のように一度intで受けてEOFかどうか判定してからcharに代入するのが正しい

```
signed char ch = EOF;
if( ch == EOF )
    printf(“end\n”);
else
    printf(“not end\n”);
```

“end”が出力されるから大丈夫？

```
int ci = getchar();
while( (ci = getchar()) != EOF){
  ch = ci;
}
```

end-of-fileを判定する

- getchar: 標準入力から 1 文字読み込み、intで返す。end-of-fileまたはエラーの場合はEOFを返す。
  - feof, ferrorを使ってend-of-fileとエラーの区別をしないといけない
- gets(char *str): 標準入力から'\n'まで文字列を読み込み、strに読み込んだ文字列へのポインタを格納する。end-of-fileまたはエラーの場合はNULLを返す。
- scanf(const char *restrict format, ...): フォーマットに従って標準入力の文字列を走査し、対応する変数に格納する。格納された変数の数が返される。end-of-fileまたはエラーの場合はEOFを返す。
- read(int fildes, void *buf, size_t nbyte): バイト数nbyteのデータをfildesから読み込み、ポインタbufで示されたバッファに格納する。成功すれば実際に読み込まれたバイト数を返す。end-of-fileの場合は0を返す。それ以外は-1を返す。

end-of-fileのキャンセル

- 一度end-of-fileを入力すると、それ以降何を入力してもプログラムではend-of-fileしか読み込めない
- clearerr関数でend-of-fileをキャンセルできる
- 標準入力から入力されたend-of-fileをキャンセルしたい場合はclearerr(stdin)

まとめ

- end-of-fileはシステム依存の整数値
- EOFマクロで値が定義されている
- 判定方法はいくつかあるが、EOFは整数値、char型は1byteであることに注意
  - ただし環境によって異なる場合がある（捕捉を参照）

捕捉

- char 型が2byte以上であり、 int 型が char 型と同じサイズであるような処理系では、getcharなどの返す値がEOFかどうか区別できないことがある

- このような場合はfeof関数を使ってEOFを判断する