6 章_その他 - 落ち穂拾い

6-1 新しい関数群

ANSI C より後の C ではセキュリティを高めるために境界チェックの機能をつけた関数や、静的な記憶領域を使わないように改善された関数がある。

6-1-1 範囲チェックが追加された関数(C11)

C では配列の範囲を超えた書き込みを行うと、バッファオーバーフロー脆弱性というセキュリティホールになりうる。 C11 では `gets` がなくなり、かわりに `gets_s` が用意された。

6-1-2 静的な領域を使わないようにした関数(C11)

C11 では `strtok` を廃止し `strtok_s` が用意された。

6-2 落とし穴

6-2-1 整数拡張

標準入力から 1 文字読み込む関数 `getchar()` の戻り値の型は int である。 C では整数拡張という機能により、int より小さなサイズの型は式の中で片っぱしから int に拡張される。

6-2-2 古い C で float 型の引数を使ったら

ANSI C 以前の C では、式の中の float 型は片っぱしから double に変換される。

6-2-3 printf() と scanf()

`printf()` のような可変長引数を取る関数においては、可変部の引数についてはプロトタイプ宣言が効かない。 よって、そのような部分については ANSI C 以前の C と同様の変換が入る。つまり整数拡張や float→double の変換が行われる。

6-2-4 プロトタイプ宣言の光と闇

- 関数を定義しているソースファイルでは、その関数自体のプロトタイプ宣言を含むヘッダファイルを必ず #include すること。
- 別ファイルで定義された関数を呼ぶ際には、必ずプロトタイプ宣言を含むヘッダファイルを #include すること。

6-3 イディオム

6-3-1 構造体宣言

構造体の書き方の派閥についての説明。

6-3-2 自己参照構造体

宣言している型と同じ型へのポインタを含む構造体を「自己参照構造体」と呼ぶことがある。 現場で聞いたことがない。

6-3-3 構造体の相互参照

タグだけ宣言した場合、そのかたは「不完全型」になる。 不完全型にはポインタを取れない。

6-3-4 ~ 6-3-7 は省略

6-3-8 char へのポインタの配列の初期化

配列の初期化子の最後の要素の後ろにはコンマをつけてもつけなくても良い

6-3-9 ~ 6-3-12 も省略