

SHC/C++ V5.1 R0xリリースノート

平成12年9月11日

下記の通り、SHC/C++コンパイラを提供いたします。

1．提供物件

PC版 SHC/C++コンパイラ Ver.5.1R13

2．不具合対策

PC版：SH C/C++ V5.1R12から、以下の不具合を対策しました。

- (1) C++で_errno,_HUGE_VALがMultiple Linkage-specificationになるケースがある。
- (2) FMACを使用するコードでFR0を壊した不正なオブジェクトを生成するケースがある。
- (3) 外部変数の宣言がなくてもコンパイルエラーにならないケースがある。

以上

SHC/C++ V5.1 R0xリリースノート

平成12年7月10日

下記の通り、SHC/C++コンパイラを提供いたします。

1．提供物件

PC版 SHC/C++コンパイラ Ver.5.1R12

2．不具合対策

PC版：SH C/C++ V5.1R11から、以下の不具合を対策しました。

- (1) if文等の条件式の直後のブロックが空で、かつその条件式が定数との判定である場合、4099インターナルエラーになる。

以上

SHC/C++ V5.1 R0xリリースノート

平成12年2月14日

下記の通り、SHC/C++コンパイラを提供いたします。

1. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R11
(Ver5.1R10は欠番です)

2. 不具合対策

PC版：SH C/C++ V5.1R09から、以下の不具合を対策しました。

- (1) -case=tableオプションで switch文のcaseラベルのコード出力方式をcaseジャンプテーブル展開するよう指定したにもかかわらずif文展開する場合がある。

【例】

```
int f(int x)
{ switch(x){
  case 0: return x;
}
}
```

- (2) ループ内で最適化によって作成された後置++式がオブジェクト不正になる場合がある

【例】

```
#define WIN_MAX 6
static int win_pos[WIN_MAX][2];
#pragma global_register(a=R8,b=R9,c=R10,d=R11,e=R12,f=R13,g=R14)
int a,b,c,d,e,f,g;

void func()
{
  int i,wn,wdn=0,x,y;
  int win_disp[WIN_MAX];
  for (i=0 ; i<WIN_MAX ; i++) {
    if (win_pos[i][0] == x && win_pos[i][1] == y ) {
      win_disp[i] = 1 ;
      wdn++ ;
    }
    else
      win_disp[i] = 0 ;
  }
}
```

- (3) ヘッドファイルmath.hとstdlib.hで宣言されているHUGE_VALの名前が一致しない。
stdlib.h 内で宣言している名前を下記の通り変更しました。

誤: extern const double HUGE_VAL;
正: extern const double _HUGE_VAL;

- (4) デストラクタを持ったクラスメンバを持つクラスがあり、そのクラスのデストラクタで明示的にreturn文を記述するとデストラクタがコールされない場合がある。

【例】

```
extern "C" int printf(const char*, ...);
struct CTestMember{
  ~CTestMember() {printf("CTestMember::~~CTestMember\n"); }
};
struct CTest{
  CTestMember ctm;
  ~CTest() { if (1) return; }
```

```
};
main()
{
    CTest test;
}
```

- (5) メンバへのポインタを static データメンバとして宣言し、その初期値として同じクラスの private メンバを指定すると不当にエラーを表示する。

【例】

```
class A{
public:
    typedef int(A::*MF)(void);
private:
    static MF na;
    int func(void);
};
```

```
A::MF A::na = &A::func; // エラー6292を不正に出力
```

- (6) メンバ関数内で構造体の集合体形式{}による初期化または static 変数の初期化を行う場合で、初期値にメンバ関数の属するクラスのメンバを指定するとインターナルエラーが出力される。

【例】

```
struct A {
    void* p;
};
struct B{
    void f();
    char a[5];
};
void B::f()
{
    A xx = {a}; // インターナルエラー
}
```

以上

平成11年9月22日

下記の通り、SHC/C++コンパイラを提供いたします。

1. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R09

2. 不具合対策

PC版：SHC/C++ V5.1R08から、以下の不具合を対策しました。

(1) 最適化指定時、レジスタ割付不正になる場合がある

例

```
extern struct A{...} *a[4], *b[4];
void foo(void)
{ short i;
  for (l=0; l<4; l++){
    *(a[l]) = *(b[l]);
  }
}
```

(2) switch文内で不当にオブジェクトのデストラクタコールする場合がある

例

```
class A {public: A(const char*){} ~A(){} };
void set_A(const A&) { }
main()
{
  int x=0;
  switch (x){
    case 1: set_A("abc"); break;
    default: break;          // ここで 不当にA::~~A()をコール
  }
}
```

(3) continue文のあるfor, whileループで帰納変数削除の最適化不正となる場合がある (-size指定時)

例

```
static void LocalFunc(void)
{
  long i,param;
  for (l=0; l<1; l++){
    param=array_param[l];
    if (param <0 ){
      continue;
    }
    GlobalFunc(param);
  }
}
```

(4) C++で不当に6203エラーを出力する

(5) tan(3/4)が、-0.0になる

(6) 入力値が大きな数のときatan, atan2の値が不正

(7) 入力値が大きな数のときsin/cosの値が-1 ~ 1の範囲外の値になる

(8) 関数ld_extで、delayed slot最適化不正となる場合がある

以上

平成11年7月19日

下記の通り、SHC/C++コンパイラを提供いたします。

2. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R08

Indy版 SHC/C++コンパイラ Ver.5.1R08

2. 不具合対策

PC版：SH C/C++ V5.1R06から、以下の不具合を対策しました。

Indy版：V5.1R02から以下の不具合までを対策しました。

(1) 関数引数の暗黙の型変換でユーザ定義のコピーコンストラクタが
Callされていない。

(2) bool型を含むEC++libraryのエンコード名不正

例

ライブラリの string クラスを使用していると、リンク時に
string::_ec2p_getmem(const char*, unsigned long, char, bool)
がないというエラーになる。

注意事項

- (1) コンパイラはatan, atan2等数学関数でのゼロ割チェックをおこないません。CPUの
ゼロ割割り込みがenableの場合はゼロ割例外を発生する可能性があります。
- (2) Ver.5.0台および Ver.5.1R01のC++をお使いの方で、Ver.5.1R08お使いになる場合は
以前のバージョンで作成したオブジェクトプログラムをリコンパイルしていただきたく、
お願い申し上げます。(C++クラス継承時のデータalignmentが最適に配置されるよう
変更されています)

以上

平成11年6月25日

下記の通り、SHC/C++コンパイラを提供いたします。

1. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R06

2. 不具合対策

SH C/C++ V5.1R05から、以下の不具合を対策しました。

(1) コピーサイズが65536以上でmemcpyライブラリの動作が不正になる場合がある。

例

```
#include <string.h>
short a[32769], b[32769];
f()
{
    memcpy(a, b, 65538);
}
```

(2) 仮想基底クラスが、2レベル以上の仮想基底クラスでない基底クラスを持つ場合のコンストラクタの動作が不正になる。

(3) switch文本体のオブジェクトサイズが32KB以上のとき、動作が不正になる場合がある。

例

```
switch (x) {           // ここから
case 0:
    ....
case 1:                // オブジェクトサイズが32KB以上
    ...
    ...
}                       // ここまで
```

注意事項

Ver.5.0台および Ver.5.1R01のC++をお使いの方で、Ver.5.1R06をお使いになる場合は以前のバージョンで作成したオブジェクトプログラムをリコンパイルしていただきたく、お願い申し上げます。(C++クラス継承時のデータalignmentが最適に配置されるよう変更されています)

以上

平成11年6月15日

SHC/C++コンパイラVer.5.1R05提供の件

下記の通り、SHC/C++コンパイラを提供いたします。

3. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R05

2. 不具合対策

SH C/C++ V5.1R04から、以下の不具合を対策しました。

- (1) 帰納変数削除最適化(定数-帰納変数のパターンが複数ある場合)の不具合を対策

例

```
void foo(int);
void SetSpriteX()
{
    int i;
    for (i=0; i<2; i++){
        foo(320-(140*(1-i*2)));
    }
}
```

- (2) クラス配列のnewでメモリが確保できない場合、実行時エラーになる。

例

```
A *ap=A[5]; // newでメモリ確保できなかった場合実行時エラー
```

- (3) C++で構造体メンバの構造体型データをバイト境界と扱うケースがあり、構造体サイズ不正となることがある。

例

```
struct A {int a;, int b;};
struct S {struct A aa; short s;};
struct S ss; // 誤:サイズが10バイト
```

- (4) C++でキャスト付きの初期値の値が不正。

例

```
int Buffer[0x4000];
unsigned int SCRATCH_END = ((unsigned int)&Buffer[0x4000/4]);
```

- (5) bool型のエンコード名が不正

```
void f(bool) -> 誤: __func__F__oo 正: __func__Fb
```

注意事項

Ver.5.0台および Ver.5.1R01のC++をお使いの方で、Ver.5.1R05をお使いになる場合は以前のバージョンで作成したオブジェクトプログラムをリコンパイルしていただきたく、お願い申し上げます。(C++クラス継承時のデータalignmentが最適に配置されるよう変更されています)

以上

平成11年4月20日

SHC/C++コンパイラVer.5.1R04提供の件

下記の通り、SHC/C++コンパイラを提供いたします。

4. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R04

2. 不具合対策

SH C/C++ V5.1R03から、以下の不具合を対策しました。

- (1) FMACコード展開時のレジスタ割り付け不正
- (2) C++構造体代入時のコード不正

注意事項

Ver.5.0台および Ver.5.1R01のC++をお使いの方で、Ver.5.1R04をお使いになる場合は以前のバージョンで作成したオブジェクトプログラムをリコンパイルしていただきたく、お願い申し上げます。(C++クラス継承時のデータalignmentが最適に配置されるよう変更されています)

以上

平成11年2月22日

SHC/C++コンパイラVer.5.1R03提供の件

下記の通り、SHC/C++コンパイラを提供いたします。

5. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R03

2. 不具合対策

SH C/C++ V5.1R01から、以下の不具合を対策しました。

- (1) FMAC を使用するコード展開においてレジスタが足りなくなったとき
4099 インターナルエラー出力
- (2) 配列の初期値に大量の文字列が指定されているときメモリアーバフロー。
- (3) C++ で `char* cp = &*(char *)0xB804000;` のようなコーディングがあるとき
4099 インターナルエラー出力
- (4) 組み込み関数の引数評価式でワークレジスタの退避がおきるとき、
オブジェクト不正になる場合がある。
- (5) `setjmp` で FR3 と FR5 の退避順序が逆になる。
- (6) `setjmp_a` で FR レジスタの退避領域が不正。

3. 仕様追加

- (1) C++ `bool`をサポートしました。
- (2) `enum`メンバの最後の`,`(カンマ)をエラーから除外しました。
- (3) 浮動小数点初期値の桁数が多い場合をエラーから警告に変更しました。
- (4) `switch`文のコード展開方式をコンパイルオプションで指定できるようにしました。
(`if then` 方式、`テーブル方式`)
- (5) ビットフィールドの型に`enum`を許すようにしました。

以上

平成10年11月24日

SHC/C++コンパイラVer.5.1R01提供の件

下記の通り、SHC/C++コンパイラを提供いたします。

6. 提供物件

PC版 SHC/C++コンパイラ Ver.5.1R01

2. 仕様追加 / 変更

SH C/C++ V5.0R32から、以下の機能が追加されました。

- (1) C プログラム内で、// コメントを使えるようになりました。
- (2) コマンドライン長の制限を 128から4096バイトに拡張しました。
- (3) ファイル名の長さを128バイトから251バイトに拡張しました。
- (4) #pragma alingdata32をサポートしました。
- (5) 環境変数 SHC_INCで指定するパス名の区切りに"."を使えるようになりました。
- (6) コンパイラのコピーライト表示をデフォルトで表示しないように変更しました。
コピーライト表示する場合は、オプション -logoを指定します。
- (7) 標準インクルードファイルを2重にインクルードすることができるようになりました。
- (8) -1.0の定数をリテラルプールからロードせず命令から生成します。
- (9) 文字列の長さを512文字から32767文字へ拡張しました。

以上