PLAYSTATION(R)3 Programmer Tool Toolchain 102.002

The following are known bugs and limitations on PLAYSTATION(R)3 Programmer Tool Toolchain package.

New updates have the release version of this package.

GCC

<PPU/SPU>

- When ppu-lv2-strip or spu-lv2-strip is in use, the following warning message may be displayed on the console.

```
$ ppu-lv2-strip -s -o test_strip.elf test.elf
BFD: test.elf: warning: Empty loadable segment detected,
                is this intentional ?
BFD: test_strip.elf: warning: allocated section `.tbss' not in segment
```

  Note that the behavior of the ELF file that is generated by this command will have no problem even when this warning message is output.

- The size of the precompiled header that the Compiler for Windows can generate is 128M bytes or less.

- When the C++ source files include 3 header files with the headers precompiled by g++ and the dependency file is created by specifying the -MMD option, only the first header described within the source files will not be written into the dependency file as follows.

  Note that "make" may continue failing to detect header modifications if invalid dependency files are once created, because compilation and dependency file generation are performed simultaneously with the -MMD option, and "make" refers to the existing dependency files. In this case, discard the dependency files before performing "make".

```
--- Source Files(test.cpp) ---
#include "test1.h"
#include "test2.h"
#include "test3.h"

--- Compiling ---
$ ppu-lv2-g++ test1.h
$ ppu-lv2-g++ test2.h
$ ppu-lv2-g++ test3.h
$ ppu-lv2-g++ -MMD -c test.cpp
```

  Note that this problem may be avoided when -O1 or more is specified as an optimization option of the compiler as follows:

```
$ ppu-lv2-g++ -MMD -O1 -c test.cpp
```

- If vmx2spu.h or spu2vmx.h is used from C programs, errors are displayed. To avoid the problem, please use them from C++ programs.

- In a vector literal with (), such as (vector signed int)(x), an error is not reported even when those other than constants or constant expressions (such as variables) are used.

- Vector literals that use {} do not operate as specified in the specifications if the elements within {} are omitted like (vector signed int){1}.
  Please specify all of the elements, like (vector signed int){1,2,3,4}.

- In Windows environment, if the -frepo option is specified and a program such as the code of the example is compiled and linked, a linking error will occur.

  Example)

```
    template<typename T>
       T Max(T a, T b)
              {
                return a > b ? a : b;
              }

    int main(){
                char cbuf;
```

```
                cbuf = Max('1', '2');
              }
```

- There is a problem that the class type and the structure type of
  an object is not aligned as specified with __attribute__((align)).
  In such a case, please specify the alignment of the first members of
  the class type and the structure type. Depending on the problem,
  it may be possible to avoid it.

  Code with the problem:
```
    template<class T> class test
    {
        public:
                T a[4];
                T b[4];
                T c[4];
     };
    typedef test<float> __attribute__ ((aligned (16))) align_test;
```

   Code in which the problem is avoided:
```
     template<class T> class test
     {
         public:
                  __attribute__ ((aligned (16))) T a[4];
                  T b[4];
                  T c[4];
     };
     typedef test<float> align_test;
```

<PPU>
(Added in Release 102.002)
- When a program that performs a call that uses a function pointer or
  a virtual member function call is compiled with a -O2 or more
  optimization option and -ffast-math, a wrong code may be generated.

- In PPU program, even when "align" is specified to the variable assigned
  to the TLS area, the alignment process may not be performed.
  This behavior is defined by the specifications of Cell OS Lv-2.

- When calling other functions from the PPU program in assembly
  language, the following should be noted to set the correct value
  for TOC pointer(r2).

  (1) Always insert nop instruction after bl instruction. By doing so,
      when the function to be called uses a different TOC pointer,
      the linker will generate a linkage code to replace nop with an
      appropriate instruction.

      Example:
```
                bl              .foo
                nop
```

  (2) Always include ".toc" section (can be empty) in the assembler
      source file. This is the current restriction on the linker.

      Example: Put the following at the beginning of the file
```
           .section        ".toc","aw"
           .section        ".text"
```

<SPU>
(Added in Release 102.002)
- Internal compiler error will occure when the SPU program
   that uses vector type is compiled with -Os option.

- When -q and --gc-sections are used at the same time as a linker option
  of SPU, -q option will be enabled but not --gc-sections option.
  This is the specifications of GNU LD.

- If position independent code (PIC) is generated using -fpic option,
  C++ virtual functions cannot be used.

- According to chapter 7 of "Cell OS Lv-2 C/C++ Language Extensions
  for SPU", when the type of an operand within an expression is double
  and the operand is a denormal number, it should be considered as 0.0
  or -0.0. However, with the current compiler, the optimization result
  may not adhere to this.

- If the -funroll-loops option and the -frerun-cse-before-sched option
  are specified at the same time, an incorrect code is generated.

Please do not specify the -frerun-cse-before-sched option.

- If an object or elf that includes the brsl instruction is disassembled with spu-lv2-objdump, not the value obtained by adding s18 and $pc, but the address that is obtained by adding s18 and $pc is displayed with the "brsl rt, s18" instruction.

- With spu-lv2-gcc and spu-lv2-g++, when the optimization option is -O1 or more and the arguments of the spu_roundf and si_frds intrinsic functions can be calculated at the time of compilation, the specified rounding mode (by default, round to nearest) is not used and the calculation is done by rounding down.

  Example)
```
vec_float f(void) {
  vec_float4 v1=spu_roundtf(((vec_double2){
    0,12345678.9
  }));
}
```

GDB

<Windows/Linux>
(Added in Release 102.002)
- When ppu-lv2-gdb/spu-lv2-gdb is booted by a user directly, the argument "info-spurs" may not be passed due to script description.
  Rewrite gdbinit.ppu or gdbinit.spu before use as shown below:

  1. Search "define info-spurs"
  2. Rewrite "monitor spursinfo ¥$arg0" that is described in the next line of the sercheed line as "monitor spursinfo $arg0"

- When debugging a program, in which multiple PPU threads exist, using bedbg, the message "watch dog timeout" may be displayed on GDB.

- When info-spurs command is executed on GDB, a file information "File not found" will be always displayed for spurs_kernel.elf or stripped elf.

- When SPU threads are generated repeatedly in the PPU program booted using bedbg -c option, execution of each SPU threads may not be restarted.

- When data, which is created with encoding enabled by the Disc Image Generator, is written into a disc, it may not be read correctly.
  For details of this bug, see the Technical Note 200609-09
  https://ps3.scedev.net/technotes/view/306/1

- When a null character ("") is contained in the string that is passed as an argument(arg/env) during execution, arg/env may not be recognized correctly from the subsequent use of arg/env.

<Windows>
- The following 4 commands that are used to debug a Raw SPU by bedbg are not available.

    rawspustoplist, rawspuid, rawspuinfo, lsaddr

  For details of this bug, see the Technical Note 200610-22.
  https://ps3.scedev.net/technotes/view/345/1