Lesson 7
SPU Runtime System (SPURS)

Playstation 3 Development

Sam Serrels and Benjamin Kenwright[1]

Abstract
TODO

Keywords
Sony, PS3, PlayStation, Setup, Windows, Target Manager, ELF, PPU, SPU, Programming, ProDG, Visual Studio, Memory alignment

[1] *Edinburgh Napier University, School of Computer Science, United Kingdom*: b.kenwright@napier.ac.uk

Contents

Preface

About the Edinburgh Napier University Game Technology Playstation 3 Development Lessons Edinburgh Napier University Game Technology Lab is one of the leading game teaching and research groups in the UK - offering students cutting edge facilities that include Sony's commercial development kits. Furthermore, within the Edinburgh Napier Game Technology group are experienced developers to assist those students aspiring to releasing their own games for PlayStation. Students have constant access to he Sony DevKits and encourage enthusiastic students to design and build their own games and applications during their spare time [1].

Previous Tutorials This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

1. Introduction

Spurs Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

tasks This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

Task Sets This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

Jobs This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

Job Chains This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

Job queues This tutorial assumes you have read the previous tutorial on compiling and deploying applications to the PS3. This tutorial will cover starting a PS3 program from scratch rather than opening a sample project.

1.1 Project Setup

As with the previous threading tutorial, some project configuration is needed to run both the PPU and SPU projects.

PPU project For the PPU project, choose ”PS3 PPU Project” from the new project wizard. At the second stage of the project wizard, choose the default ”PPU ELF Project” as the project type and the SN systems compiler. Uncheck the ”Use Precompiled Headers” options.

SPU project For the SPU project, choose ”PS3 SPU Project” from the new project wizard, Fig: 1.

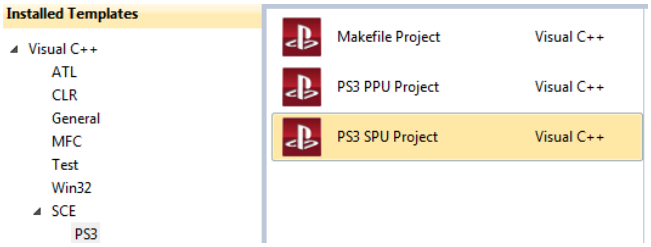

Figure 1. SPU project wizard stage 1 -

In the second stage of the project wizard, choose ”SPU ELF project” as the project type and ”SPURS Task” as the SPURS usage, Fig: 2.

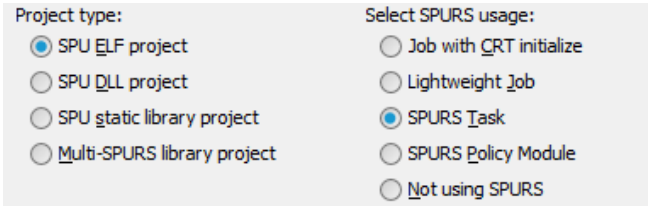

Figure 2. SPU project wizard stage 2 -

Dependency Add the SPU project as a dependency to the PPU project (the PPU project depends on the SPU project, not vice versa), Fig: 3.

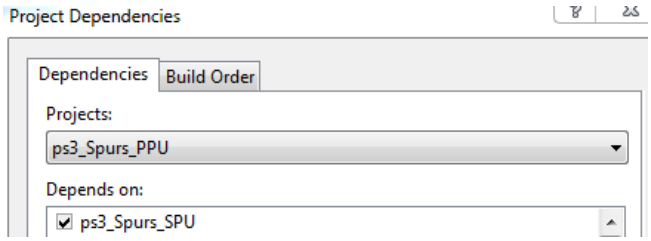

Figure 3. PPU project dependency -

Project Reference Add the SPU project as a reference to the PPU project, Fig: 4.

Embedding the SPU Task Due to the SPURS project option chosen in the new project wizard, when the SPU project is compiled it will get inserted into the PPU project in the link stage. The PPU program can reference this using an External CellSpursTaskBinInfo object, however the naming of the object is critical.

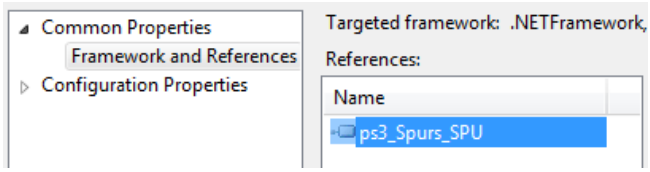

Figure 4. SPU project reference -

```
extern const CellSpursTaskBinInfo ↩
    _binary_task_ProjectName_elf_taskbininfo;
```

The name depends on the SPU project name, for example, if the SPU project was called ”ps3_Spurs_SPU”, the extern would be:

```
extern const CellSpursTaskBinInfo ↩
    _binary_task_ps3_Spurs_SPU_elf_taskbininfo;
```

The compiled SPURS task could be loaded in like a shader if it is preferred, however having it embedded directly in the code at compile time means you don’t have to deal with the file-system and the errors associated with it.

1.2 Bbbbb

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Figure 5. Memory structure -

Aaaa Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.3 Aaa

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

bb Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Listing 1. Halts on different platforms

```
1 //IA-32 (Intel Architecture, 32-bit)
2 __asm { int 3 }
3 //x86/XBOX/Win32(basically a robust wrapper for int 3)
4 //Only supported in visual studio
5 __debugbreak() ;
6 // Halts a program running on PPC32 or PPC64 (e.g. PS3).
7 //Also works for ARM and in GCC/XCode
8 __asm volatile( "trap" ) ;
```

2. Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Recommended Reading

Programming the Cell Processor: For Games, Graphics, and Computation, Matthew Scarpino, ISBN: 978-0136008866
Vector Games Math Processors (Wordware Game Math Library), James Leiterman, ISBN: 978-1556229213
Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, ISBN: 978-0132350884

References

[1] Edinburgh Napier Game Technology Website. www.napier.ac.uk/games/. *Accessed: Feb 2014*, 2014. 1