

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Converting a regular kernel into MOOL kernel

A brief overview

Pradeepkumar Gayam

DOS Lab, IIT Madras

June 30, 2018

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

1 Objective

2 Challenges

3 C++ support :: Runtime libraries

4 C++ support :: Keywords

5 C++ support :: Ctors for LKM

6 C++ support :: Language incompatibilities

Objective

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

..is to be able to write C++ code in Linux kernel.

How?

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- C & C++ are being used together from a longtime. Multiple solutions are available.

How?

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- C & C++ are being used together from a longtime. Multiple solutions are available.
- These solutions are relatively easy to implement.

How?

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- C & C++ are being used together from a longtime. Multiple solutions are available.
- These solutions are relatively easy to implement.
- There are two common approaches.
 - Compile the whole codebase with C++ compiler and fix the incompatibilities.
 - Compile different sources with their corresponding compilers and combine the object files at link time.

How?

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- C & C++ are being used together from a longtime. Multiple solutions are available.
- These solutions are relatively easy to implement.
- There are two common approaches.
 - Compile the whole codebase with C++ compiler and fix the incompatibilities.
 - Compile different sources with their corresponding compilers and combine the object files at link time.
- We have taken the second approach.

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

① Objective

② Challenges

③ C++ support :: Runtime libraries

④ C++ support :: Keywords

⑤ C++ support :: Ctors for LKM

⑥ C++ support :: Language incompatibilities

User space vs Kernel space

Conversion to
MOOL kernel

Pradeepkumar
Gayam

- Freestanding vs Hosted compilers

```
# Make variables (CC, etc...)
```

```
AS      = $(CROSS_COMPILE)as
LD      = $(CROSS_COMPILE)ld -nostdlib -nodefaultlibs -nostartfiles
REAL_CC = $(CROSS_COMPILE)gcc
CXX     = $(CROSS_COMPILE)g++ $(KBUILD_CFLAGS) $(KBUILD_CXXFLAGS)
CPP     = $(CC) -E
AR      = $(CROSS_COMPILE)ar
NM      = $(CROSS_COMPILE)nm
STRIP   = $(CROSS_COMPILE)strip
OBJCOPY = $(CROSS_COMPILE)objcopy
OBJDUMP = $(CROSS_COMPILE)objdump
AWK     = awk
```

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

User space vs Kernel space

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Freestanding vs Hosted compilers

```
# Make variables (CC, etc...)
AS      = $(CROSS_COMPILE)as
LD      = $(CROSS_COMPILE)ld -nostdlib -nodefaultlibs -nostartfiles
REAL_CC = $(CROSS_COMPILE)gcc
CXX     = $(CROSS_COMPILE)g++ $(KBUILD_CFLAGS) $(KBUILD_CXXFLAGS)
CPP     = $(CC) -E
AR      = $(CROSS_COMPILE)ar
NM      = $(CROSS_COMPILE)nm
STRIP   = $(CROSS_COMPILE)strip
OBJCOPY = $(CROSS_COMPILE)objcopy
OBJDUMP = $(CROSS_COMPILE)objdump
AWK     = awk
```

- These libraries provide runtime support for C++

Challenges

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Unavailability of runtime libraries

Challenges

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Unavailability of runtime libraries
- Keywords

Challenges

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Unavailability of runtime libraries
- Keywords
- ctors for loadable kernel modules

Challenges

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Unavailability of runtime libraries
- Keywords
- ctors for loadable kernel modules
- Language incompatibilities

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

① Objective

② Challenges

③ C++ support :: Runtime libraries

④ C++ support :: Keywords

⑤ C++ support :: Ctors for LKM

⑥ C++ support :: Language incompatibilities

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(new, delete)

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(new, delete)
- Global constructors/destructors

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(new, delete)
- Global constructors/destructors
- Virtual functions

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(new, delete)
- Global constructors/destructors
- Virtual functions
- Exceptions

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(new, delete)
- Global constructors/destructors
- Virtual functions
- Exceptions
- Dynamic type checking

Runtime libraries

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Runtime libraries provide support for the following features.

- Memory allocation operators(`new`, `delete`)
- Global constructors/destructors
- Virtual functions
- Exceptions
- Dynamic type checking

Fix : Move the relevant parts of `libgcc` & `libstdc++` into the kernel.

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

① Objective

② Challenges

③ C++ support :: Runtime libraries

④ C++ support :: Keywords

⑤ C++ support :: Ctors for LKM

⑥ C++ support :: Language incompatibilities

Keywords

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Linux kernel is written in C

Keywords

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Linux kernel is written in C
- C++ source files do need to include C files and these C files use C++ keywords as variable names.

Keywords

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- Linux kernel is written in C
- C++ source files do need to include C files and these C files use C++ keywords as variable names.

Fix : Redefine the identifiers to names accepted by the C++ compiler

```
#define new newx
#define namespace namespacex
#define private privatex
#define class classx
#define typename typenamex
#define virtual virtualx
```

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

① Objective

② Challenges

③ C++ support :: Runtime libraries

④ C++ support :: Keywords

⑤ C++ support :: Ctors for LKM

⑥ C++ support :: Language incompatibilities

Support for C++ loadable kernel modules

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- g++ secretly links two object files at the front and the back, crtbegin.o and crtend.o. This is necessary to ensure that global constructors and destructors are run, and to enable exceptions.

Support for C++ loadable kernel modules

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- g++ secretly links two object files at the front and the back, crtbegin.o and crtend.o. This is necessary to ensure that global constructors and destructors are run, and to enable exceptions.
- g++ adds initialization code into the .init ELF section and cleanup into the .fini section.

Support for C++ loadable kernel modules

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

- g++ secretly links two object files at the front and the back, crtbegin.o and crtend.o. This is necessary to ensure that global constructors and destructors are run, and to enable exceptions.
- g++ adds initialization code into the .init ELF section and cleanup into the .fini section.
- Kernel module loader in Linux pays no attention to the ELF .init section

Support for C++ loadable kernel modules

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Fix: change the definition of module initialization functions, `module_init` and `module_exit` to call initialization routines.

```
#define module_init(initfn)
int init_module(void)
{
    begin_init();
    end_init();
    return initfn();
}
```

```
/* These are defined in lib/gcc/crtstuff.c */
extern void begin_init(void);
extern void end_init(void);
extern void begin_fini(void);
```

Table of Contents

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

① Objective

② Challenges

③ C++ support :: Runtime libraries

④ C++ support :: Keywords

⑤ C++ support :: Ctors for LKM

⑥ C++ support :: Language incompatibilities

Designated initializers

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

TODO

Questions?

Conversion to
MOOL kernel

Pradeepkumar
Gayam

Objective

Challenges

C++ support ::
Runtime
libraries

C++ support ::
Keywords

C++ support ::
Ctors for LKM

C++ support ::
Language
incompatibilities

Questions?