How Ocarina of Time was decompiled

Darius Engler

November 06, 2021

Overview

Overview

- Presentation

Overview

- Presentation
- Matching decompilation

Overview

- Presentation
- Matching decompilation
- Workflow/Tooling

Presentation

Why?

Why?

- Code documentation

Why?

- Code documentation
- Glitch hunting (speedrunning)

Why?

- Code documentation
- Glitch hunting (speedrunning)
- Modding

Why?

- Code documentation
- Glitch hunting (speedrunning)
- Modding
- Fun/Challenge

"Matching decompilation"?

"Matching decompilation"?

- Byte identical output

"Matching decompilation"?

- Byte identical output
- Harder

"Matching decompilation"?

- Byte identical output
- Harder
- Equivalence testing becomes trivial
  (we just compare the hashes)

What do we need to figure out?

What do we need to figure out?

- Which compiler was used?

What do we need to figure out?

- Which compiler was used?
- Which compilers flags were used?

IRIS Developer's Option: IDO

IRIS Developer's Option: IDO

- C compiler developed by Silicon Graphics (SGI)

IRIS Developer's Option: IDO

- C compiler developed by Silicon Graphics (SGI)
- Part of IRIX OS

IRIS Developer's Option: IDO

- C compiler developed by Silicon Graphics (SGI)
- Part of IRIX OS
- Discontinued

How to differentiate IDO from GCC?

How to differentiate IDO from GCC?

We can search for pattern differences in GCC and IDO

How to differentiate IDO from GCC?

We can search for pattern differences in GCC and IDO

```c
void test(int x) {}
```

How to differentiate IDO from GCC?

We can search for pattern differences in GCC and IDO

```
void test(int x) {}
```

IDO

```
00000000 <test>:
   0:   03e00008        jr      ra
   4:   afa40000        sw      a0,0(sp)
```

How to differentiate IDO from GCC?

We can search for pattern differences in GCC and IDO

```c
void test(int x) {}
```

IDO

```
00000000 <test>:
   0:   03e00008        jr      ra
   4:   afa40000        sw      a0,0(sp)
```

GCC

```
00000000 <test>:
   0:   03e00008        jr      ra
   4:   00000000        nop
```

In our case:

In our case:

- Most of the code compiled with IDO 7.1

In our case:

- Most of the code compiled with IDO 7.1
- Some of "libultra" compiled with IDO 5.3

Running IDO is a hell

Running IDO is a hell

IDO runs on proprietary MIPS workstation
and is closed source

Running IDO is a hell

IDO runs on proprietary MIPS workstation
and is closed source

At first we were using QEMU

Running IDO is a hell

IDO runs on proprietary MIPS workstation
and is closed source

At first we were using QEMU

Switched to static recompilation of IDO

Static Recompilation

Static Recompilation

Translates assembly to ugly but compilable C code

```
a3 = 0x0;
a3 = a3 + gp;
a3 = MEM_U32(a3 + -32684);
a0 = MEM_U32(sp + 0);
a1 = sp + 0x4;
a3 = MEM_U32(a3 + 0);
at = 0xfffffff0;
sp = sp & at;
a2 = a1 + 0x4;
v0 = a0 << 2;
sp = sp + 0xffffffe0;
if (a3 != 0) {a2 = a2 + v0;
goto L40cf80;}
a2 = a2 + v0;
at = 0x0;
at = at + gp;
at = MEM_U32(at + -32684);
MEM_U32(at + 0) = a2;
L40cf80:
at = 0x0;
at = at + gp;
at = MEM_U32(at + -30028);
```

Static Recompilation

Translates assembly to ugly but compilable C code

⇒ We get a native binary

```
a3 = 0x0;
a3 = a3 + gp;
a3 = MEM_U32(a3 + -32684);
a0 = MEM_U32(sp + 0);
a1 = sp + 0x4;
a3 = MEM_U32(a3 + 0);
at = 0xfffffff0;
sp = sp & at;
a2 = a1 + 0x4;
v0 = a0 << 2;
sp = sp + 0xffffffe0;
if (a3 != 0) {a2 = a2 + v0;
goto L40cf80;}
a2 = a2 + v0;
at = 0x0;
at = at + gp;
at = MEM_U32(at + -32684);
MEM_U32(at + 0) = a2;
L40cf80:
at = 0x0;
at = at + gp;
at = MEM_U32(at + -30028);
```

Static Recompilation

Translates assembly to ugly but compilable C code

⇒ We get a native binary

QEMU

```
real    3m17,946s
user    22m24,901s
sys     1m4,315s
```

```
a3 = 0x0;
a3 = a3 + gp;
a3 = MEM_U32(a3 + -32684);
a0 = MEM_U32(sp + 0);
a1 = sp + 0x4;
a3 = MEM_U32(a3 + 0);
at = 0xfffffff0;
sp = sp & at;
a2 = a1 + 0x4;
v0 = a0 << 2;
sp = sp + 0xffffffe0;
if (a3 != 0) {a2 = a2 + v0;
goto L40cf80;}
a2 = a2 + v0;
at = 0x0;
at = at + gp;
at = MEM_U32(at + -32684);
MEM_U32(at + 0) = a2;
L40cf80:
at = 0x0;
at = at + gp;
at = MEM_U32(at + -30028);
```

Static Recompilation

Translates assembly to ugly but compilable C code

⇒ We get a native binary

QEMU

```
real      3m17,946s
user      22m24,901s
sys       1m4,315s
```

Static Recomp

```
real      1m12,407s
user      6m45,375s
sys       0m40,974s
```

```
a3 = 0x0;
a3 = a3 + gp;
a3 = MEM_U32(a3 + -32684);
a0 = MEM_U32(sp + 0);
a1 = sp + 0x4;
a3 = MEM_U32(a3 + 0);
at = 0xfffffff0;
sp = sp & at;
a2 = a1 + 0x4;
v0 = a0 << 2;
sp = sp + 0xffffffe0;
if (a3 != 0) {a2 = a2 + v0;
goto L40cf80;}
a2 = a2 + v0;
at = 0x0;
at = at + gp;
at = MEM_U32(at + -32684);
MEM_U32(at + 0) = a2;
L40cf80:
at = 0x0;
at = at + gp;
at = MEM_U32(at + -30028);
```

Compiler flags

Compiler flags

- Most of the code compiled with -O2

Compiler flags

- Most of the code compiled with -O2
- Some of libultra compiled with -O1

Compiler flags

- Most of the code compiled with -O2
- Some of libultra compiled with -O1
- Some files compiled with -O2 -g3

Compiler flags

- Most of the code compiled with -O2
- Some of libultra compiled with -O1
- Some files compiled with -O2 -g3
- Some files compiled with -mips3 -32

Compiler flags

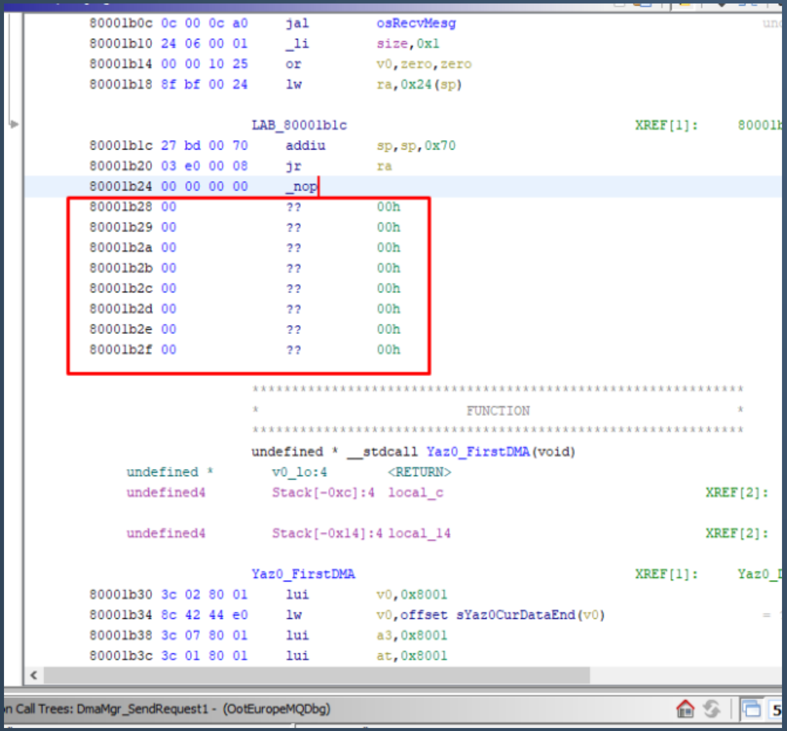
- Most of the code compiled with -O2
- Some of libultra compiled with -O1
- Some files compiled with -O2 -g3
- Some files compiled with -mips3 -32
- Some files compiled with -trapuv

Workflow/Tooling

Separating object files

Separating object files

IDO alignes every object file to 0x10

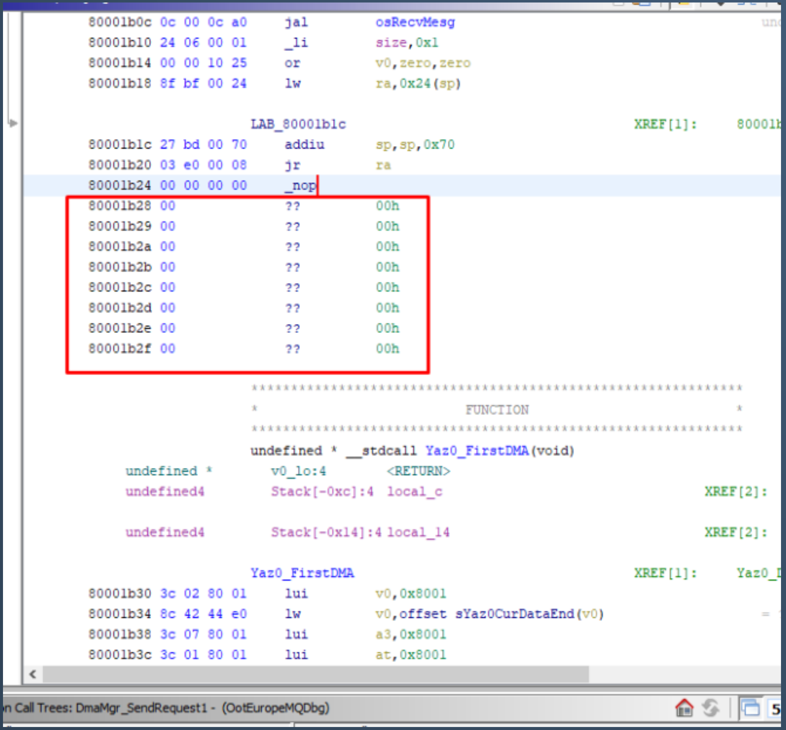


```
   80001b0c 0c 00 0c a0     jal        osRecvMesg                                    un
   80001b10 24 06 00 01     _li        size,0x1
   80001b14 00 00 10 25     or         v0,zero,zero
   80001b18 8f bf 00 24     lw         ra,0x24(sp)

                         LAB_80001b1c                                    XREF[1]:     80001b
   80001b1c 27 bd 00 70     addiu      sp,sp,0x70
   80001b20 03 e0 00 08     jr         ra
   80001b24 00 00 00 00     _nop
   80001b28 00              ??         00h
   80001b29 00              ??         00h
   80001b2a 00              ??         00h
   80001b2b 00              ??         00h
   80001b2c 00              ??         00h
   80001b2d 00              ??         00h
   80001b2e 00              ??         00h
   80001b2f 00              ??         00h

                         **************************************************************
                         *                          FUNCTION                          *
                         **************************************************************
                         undefined * __stdcall Yaz0_FirstDMA(void)
        undefined *        v0_lo:4        <RETURN>
        undefined4         Stack[-0xc]:4  local_c                                XREF[2]:

        undefined4         Stack[-0x14]:4 local_14                               XREF[2]:

                         Yaz0_FirstDMA                                   XREF[1]:     Yaz0_
   80001b30 3c 02 80 01     lui        v0,0x8001
   80001b34 8c 42 44 e0     lw         v0,offset sYaz0CurDataEnd(v0)                  =
   80001b38 3c 07 80 01     lui        a3,0x8001
   80001b3c 3c 01 80 01     lui        at,0x8001
```

n Call Trees: DmaMgr_SendRequest1 - (OotEuropeMQDbg)

Separating object files

IDO alignes every object file to 0x10

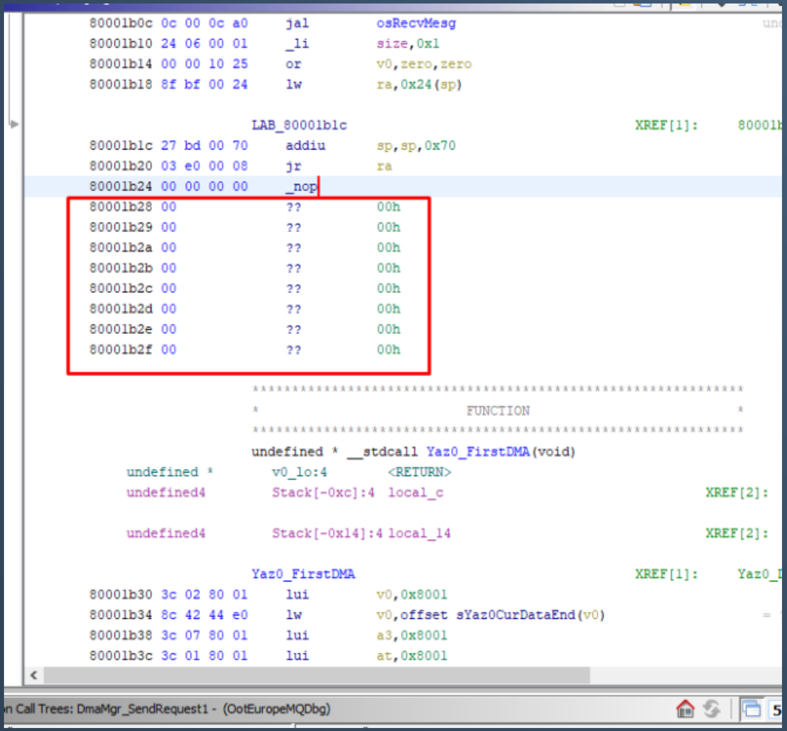


```
80001b0c 0c 00 0c a0     jal         osRecvMesg                                        un
80001b10 24 06 00 01     _li         size,0x1
80001b14 00 00 10 25     or          v0,zero,zero
80001b18 8f bf 00 24     lw          ra,0x24(sp)

                     LAB_80001b1c                                      XREF[1]:     80001b
80001b1c 27 bd 00 70     addiu       sp,sp,0x70
80001b20 03 e0 00 08     jr          ra
80001b24 00 00 00 00     _nop
80001b28 00              ??          00h
80001b29 00              ??          00h
80001b2a 00              ??          00h
80001b2b 00              ??          00h
80001b2c 00              ??          00h
80001b2d 00              ??          00h
80001b2e 00              ??          00h
80001b2f 00              ??          00h

                     **************************************************************
                     *                          FUNCTION                          *
                     **************************************************************
                     undefined * __stdcall Yaz0_FirstDMA(void)
     undefined *        v0_lo:4        <RETURN>
     undefined4         Stack[-0xc]:4  local_c                                 XREF[2]:

     undefined4         Stack[-0x14]:4 local_14                                XREF[2]:

                     Yaz0_FirstDMA                                     XREF[1]:     Yaz0_D
80001b30 3c 02 80 01     lui         v0,0x8001
80001b34 8c 42 44 e0     lw          v0,offset sYaz0CurDataEnd(v0)                     =
80001b38 3c 07 80 01     lui         a3,0x8001
80001b3c 3c 01 80 01     lui         at,0x8001
```

75% chance of detecting a file boundary automatically

Separating object files

IDO alignes every object file to 0x10



```
80001b0c 0c 00 0c a0     jal         osRecvMesg                                        und
80001b10 24 06 00 01     _li         size,0x1
80001b14 00 00 10 25     or          v0,zero,zero
80001b18 8f bf 00 24     lw          ra,0x24(sp)

                     LAB_80001b1c                                XREF[1]:     80001b
80001b1c 27 bd 00 70     addiu       sp,sp,0x70
80001b20 03 e0 00 08     jr          ra
80001b24 00 00 00 00     _nop
80001b28 00              ??          00h
80001b29 00              ??          00h
80001b2a 00              ??          00h
80001b2b 00              ??          00h
80001b2c 00              ??          00h
80001b2d 00              ??          00h
80001b2e 00              ??          00h
80001b2f 00              ??          00h

                     ***************************************************************
                     *                          FUNCTION                           *
                     ***************************************************************
                     undefined * __stdcall Yaz0_FirstDMA(void)
       undefined *       v0_lo:4        <RETURN>
       undefined4        Stack[-0xc]:4  local_c                                XREF[2]:

       undefined4        Stack[-0x14]:4 local_14                               XREF[2]:

                     Yaz0_FirstDMA                               XREF[1]:     Yaz0_I
80001b30 3c 02 80 01     lui         v0,0x8001
80001b34 8c 42 44 e0     lw          v0,offset sYaz0CurDataEnd(v0)                       =
80001b38 3c 07 80 01     lui         a3,0x8001
80001b3c 3c 01 80 01     lui         at,0x8001
```

n Call Trees: DmaMgr_SendRequest1 - (OotEuropeMQDbg)

75% chance of detecting a file boundary automatically

The rest can be guessed based on context

Decompiling a file

Decompiling a file

- Split the assembly file into one file per function

Decompiling a file

- Split the assembly file into one file per function
- Add a C file that includes all the assembly

Decompiling a file

- Split the assembly file into one file per function
- Add a C file that includes all the assembly
- Progressively turn every function from asm to C

Problem: IDO doesn't support inline assembly

Problem: IDO doesn't support inline assembly

"__asm__" not supported

Problem: IDO doesn't support inline assembly

"__asm__" not supported

"asm-processor" was written to add a custom pragma directive for that

Problem: IDO doesn't support inline assembly

"__asm__" not supported

"asm-processor" was written to add a custom pragma directive for that

```
#pragma GLOBAL_ASM("my_function.s")
```

Problem: IDO doesn't support inline assembly

"__asm__" not supported

"asm-processor" was written to add a custom pragma directive for that

```
#pragma GLOBAL_ASM("my_function.s")
```

→

```
void my_function(void) {
  *(volatile int*)0 = 0; // sw zero,0(zero)
  *(volatile int*)0 = 0; // sw zero,0(zero)
  *(volatile int*)0 = 0; // sw zero,0(zero)
  ...
}
```

Decompiler

Decompiler

mips_to_c

Decompiler

mips_to_c

- Specific to MIPS
- Recognizes IDO specific patterns
- Great for matching

Decompiler

mips_to_c

- Specific to MIPS
- Recognizes IDO specific patterns
- Great for matching
- Not interactive

Decompiler

mips_to_c

- Specific to MIPS
- Recognizes IDO specific patterns
- Great for matching
- Not interactive

Ghidra

Decompiler

mips_to_c

- Specific to MIPS
- Recognizes IDO specific patterns
- Great for matching
- Not interactive

Ghidra

- Interactive
- Great for documentation

Decompiler

mips_to_c

- Specific to MIPS
- Recognizes IDO specific patterns
- Great for matching
- Not interactive

Ghidra

- Interactive
- Great for documentation
- Not ideal for matching

Original

```
u32 StackCheck_CheckAll(void) {
    u32 ret = 0;
    StackEntry* iter = sStackInfoListStart;

    while (iter) {
        u32 state = StackCheck_GetState(iter);
        if (state != STACK_STATUS_OK) {
            ret = 1;
        }
        iter = iter->next;
    }

    return ret;
}
```

Original

```
u32 StackCheck_CheckAll(void) {
    u32 ret = 0;
    StackEntry* iter = sStackInfoListStart;

    while (iter) {
        u32 state = StackCheck_GetState(iter);
        if (state != STACK_STATUS_OK) {
            ret = 1;
        }
        iter = iter->next;
    }

    return ret;
}
```

Ghidra

```
int StackCheck_CheckAll(void)
{
    int state;
    StackEntry *iter;
    int ret;

    ret = 0;
    iter = sStackInfoListStart;
    if (sStackInfoListStart != (StackEntry *)0x0) {
        do {
            state = StackCheck_GetState((int)iter);
            if (state == 0) {
                iter = iter->next;
            }
            else {
                ret = 1;
                iter = iter->next;
            }
        } while (iter != (StackEntry *)0x0);
    }
    return ret;
}
```

Original

```c
u32 StackCheck_CheckAll(void) {
    u32 ret = 0;
    StackEntry* iter = sStackInfoListStart;

    while (iter) {
        u32 state = StackCheck_GetState(iter);
        if (state != STACK_STATUS_OK) {
            ret = 1;
        }
        iter = iter->next;
    }

    return ret;
}
```

Ghidra

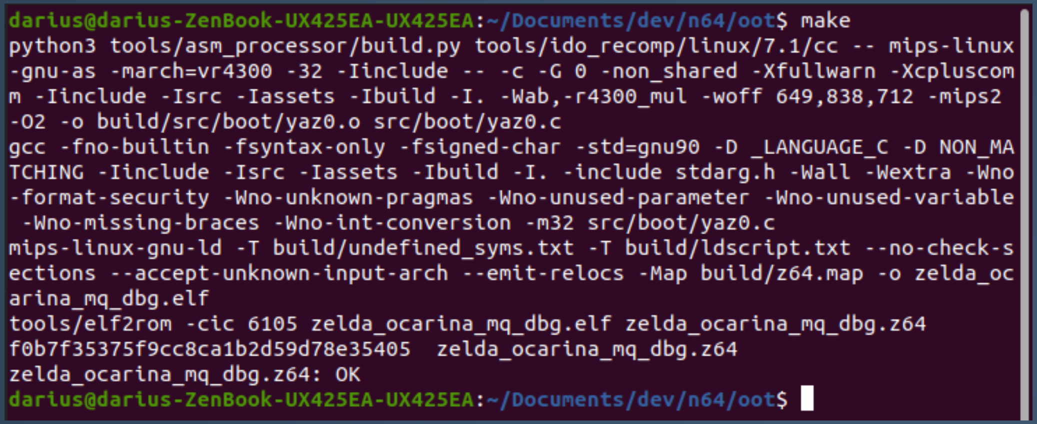
```c
int StackCheck_CheckAll(void)
{
    int state;
    StackEntry *iter;
    int ret;

    ret = 0;
    iter = sStackInfoListStart;
    if (sStackInfoListStart != (StackEntry *)0x0) {
        do {
            state = StackCheck_GetState((int)iter);
            if (state == 0) {
                iter = iter->next;
            }
            else {
                ret = 1;
                iter = iter->next;
            }
        } while (iter != (StackEntry *)0x0);
    }
    return ret;
}
```

mips2c

```c
u32 StackCheck_CheckAll(void) {
    StackEntry *temp_s0;
    StackEntry *temp_s0_2;
    StackEntry *phi_s0;
    u32 phi_s1;
    u32 phi_s1_2;

    temp_s0 = sStackInfoListStart;
    phi_s0 = temp_s0;
    phi_s1 = 0U;
    phi_s1_2 = 0U;
    if (temp_s0 != 0) {
        do {
            if (StackCheck_GetState(phi_s0) != 0) {
                phi_s1_2 = 1U;
            }
            temp_s0_2 = phi_s0->next;
            phi_s0 = temp_s0_2;
            phi_s1 = phi_s1_2;
        } while (temp_s0_2 != 0);
    }
    return phi_s1;
}
```

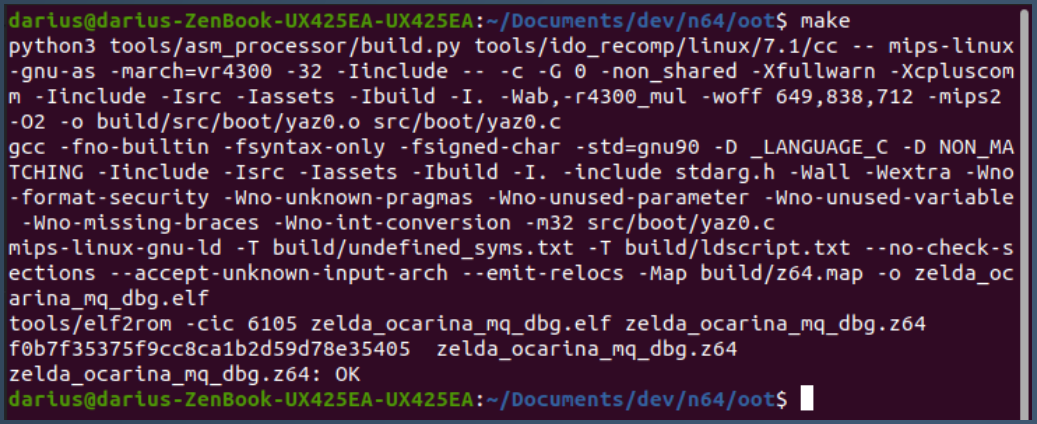
Checking for differences

Checking for differences

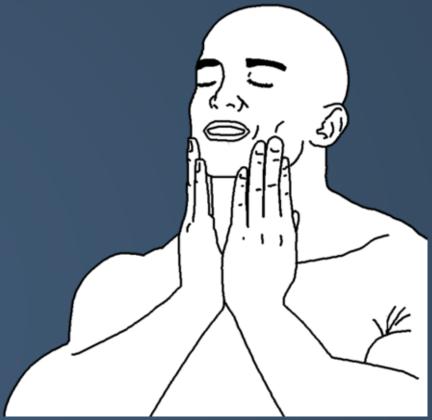


```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$ make
python3 tools/asm_processor/build.py tools/ido_recomp/linux/7.1/cc -- mips-linux
-gnu-as -march=vr4300 -32 -Iinclude -- -c -G 0 -non_shared -Xfullwarn -Xcpluscom
m -Iinclude -Isrc -Iassets -Ibuild -I. -Wab,-r4300_mul -woff 649,838,712 -mips2
-O2 -o build/src/boot/yaz0.o src/boot/yaz0.c
gcc -fno-builtin -fsyntax-only -fsigned-char -std=gnu90 -D _LANGUAGE_C -D NON_MA
TCHING -Iinclude -Isrc -Iassets -Ibuild -I. -include stdarg.h -Wall -Wextra -Wno
-format-security -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-variable
 -Wno-missing-braces -Wno-int-conversion -m32 src/boot/yaz0.c
mips-linux-gnu-ld -T build/undefined_syms.txt -T build/ldscript.txt --no-check-s
ections --accept-unknown-input-arch --emit-relocs -Map build/z64.map -o zelda_oc
arina_mq_dbg.elf
tools/elf2rom -cic 6105 zelda_ocarina_mq_dbg.elf zelda_ocarina_mq_dbg.z64
f0b7f35375f9cc8ca1b2d59d78e35405  zelda_ocarina_mq_dbg.z64
zelda_ocarina_mq_dbg.z64: OK
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$
```

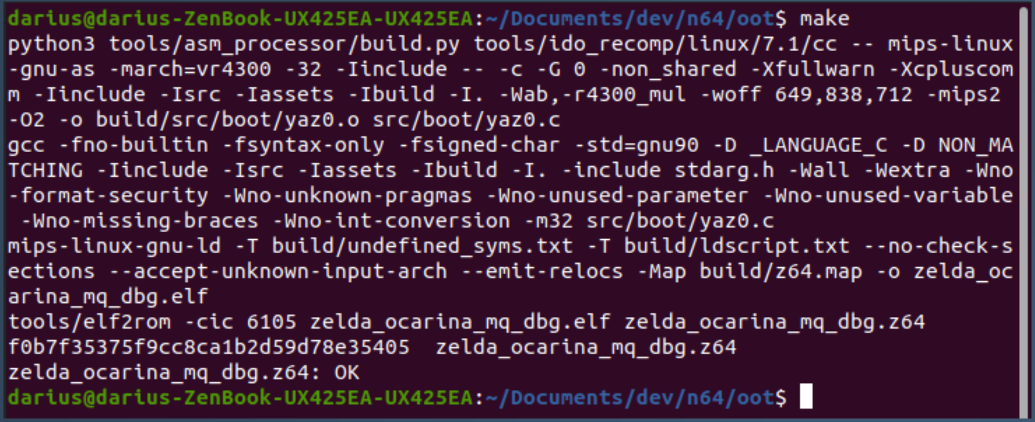
Checking for differences



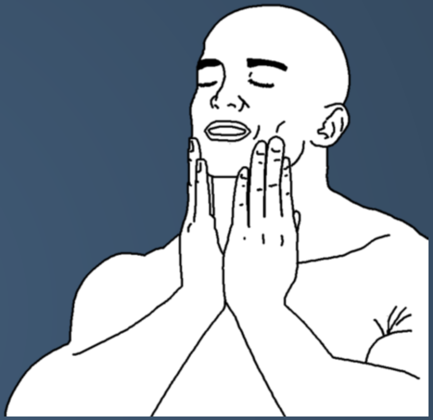
```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$ make
python3 tools/asm_processor/build.py tools/ido_recomp/linux/7.1/cc -- mips-linux
-gnu-as -march=vr4300 -32 -Iinclude -- -c -G 0 -non_shared -Xfullwarn -Xcpluscom
m -Iinclude -Isrc -Iassets -Ibuild -I. -Wab,-r4300_mul -woff 649,838,712 -mips2
-O2 -o build/src/boot/yaz0.o src/boot/yaz0.c
gcc -fno-builtin -fsyntax-only -fsigned-char -std=gnu90 -D _LANGUAGE_C -D NON_MA
TCHING -Iinclude -Isrc -Iassets -Ibuild -I. -include stdarg.h -Wall -Wextra -Wno
-format-security -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-variable
 -Wno-missing-braces -Wno-int-conversion -m32 src/boot/yaz0.c
mips-linux-gnu-ld -T build/undefined_syms.txt -T build/ldscript.txt --no-check-s
ections --accept-unknown-input-arch --emit-relocs -Map build/z64.map -o zelda_oc
arina_mq_dbg.elf
tools/elf2rom -cic 6105 zelda_ocarina_mq_dbg.elf zelda_ocarina_mq_dbg.z64
f0b7f35375f9cc8ca1b2d59d78e35405  zelda_ocarina_mq_dbg.z64
zelda_ocarina_mq_dbg.z64: OK
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$ 
```

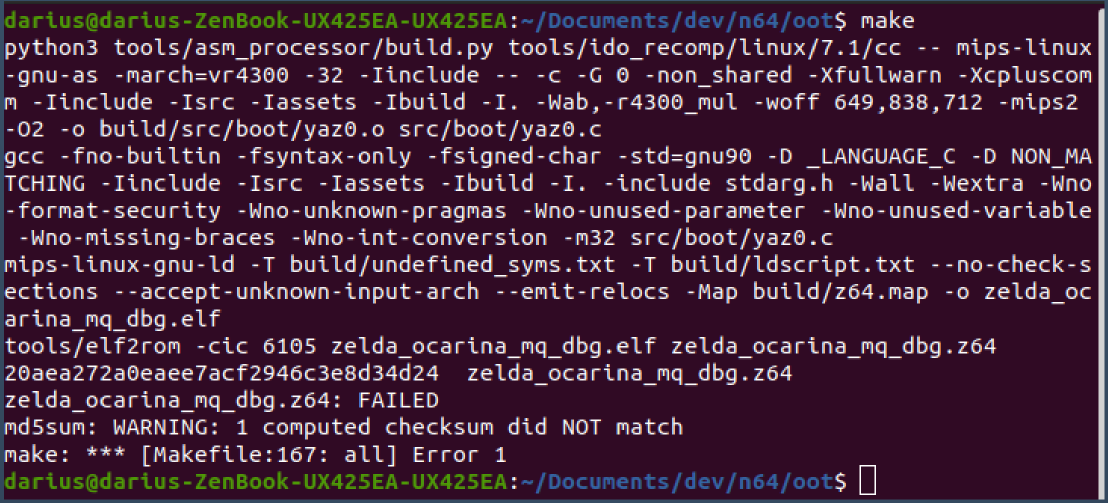


Checking for differences



```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$ make
python3 tools/asm_processor/build.py tools/ido_recomp/linux/7.1/cc -- mips-linux
-gnu-as -march=vr4300 -32 -Iinclude -- -c -G 0 -non_shared -Xfullwarn -Xcpluscom
m -Iinclude -Isrc -Iassets -Ibuild -I. -Wab,-r4300_mul -woff 649,838,712 -mips2
-O2 -o build/src/boot/yaz0.o src/boot/yaz0.c
gcc -fno-builtin -fsyntax-only -fsigned-char -std=gnu90 -D _LANGUAGE_C -D NON_MA
TCHING -Iinclude -Isrc -Iassets -Ibuild -I. -include stdarg.h -Wall -Wextra -Wno
-format-security -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-variable
 -Wno-missing-braces -Wno-int-conversion -m32 src/boot/yaz0.c
mips-linux-gnu-ld -T build/undefined_syms.txt -T build/ldscript.txt --no-check-s
ections --accept-unknown-input-arch --emit-relocs -Map build/z64.map -o zelda_oc
arina_mq_dbg.elf
tools/elf2rom -cic 6105 zelda_ocarina_mq_dbg.elf zelda_ocarina_mq_dbg.z64
f0b7f35375f9cc8ca1b2d59d78e35405  zelda_ocarina_mq_dbg.z64
zelda_ocarina_mq_dbg.z64: OK
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$
```





```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$ make
python3 tools/asm_processor/build.py tools/ido_recomp/linux/7.1/cc -- mips-linux
-gnu-as -march=vr4300 -32 -Iinclude -- -c -G 0 -non_shared -Xfullwarn -Xcpluscom
m -Iinclude -Isrc -Iassets -Ibuild -I. -Wab,-r4300_mul -woff 649,838,712 -mips2
-O2 -o build/src/boot/yaz0.o src/boot/yaz0.c
gcc -fno-builtin -fsyntax-only -fsigned-char -std=gnu90 -D _LANGUAGE_C -D NON_MA
TCHING -Iinclude -Isrc -Iassets -Ibuild -I. -include stdarg.h -Wall -Wextra -Wno
-format-security -Wno-unknown-pragmas -Wno-unused-parameter -Wno-unused-variable
 -Wno-missing-braces -Wno-int-conversion -m32 src/boot/yaz0.c
mips-linux-gnu-ld -T build/undefined_syms.txt -T build/ldscript.txt --no-check-s
ections --accept-unknown-input-arch --emit-relocs -Map build/z64.map -o zelda_oc
arina_mq_dbg.elf
tools/elf2rom -cic 6105 zelda_ocarina_mq_dbg.elf zelda_ocarina_mq_dbg.z64
20aea272a0eaee7acf2946c3e8d34d24  zelda_ocarina_mq_dbg.z64
zelda_ocarina_mq_dbg.z64: FAILED
md5sum: WARNING: 1 computed checksum did NOT match
make: *** [Makefile:167: all] Error 1
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/oot$
```

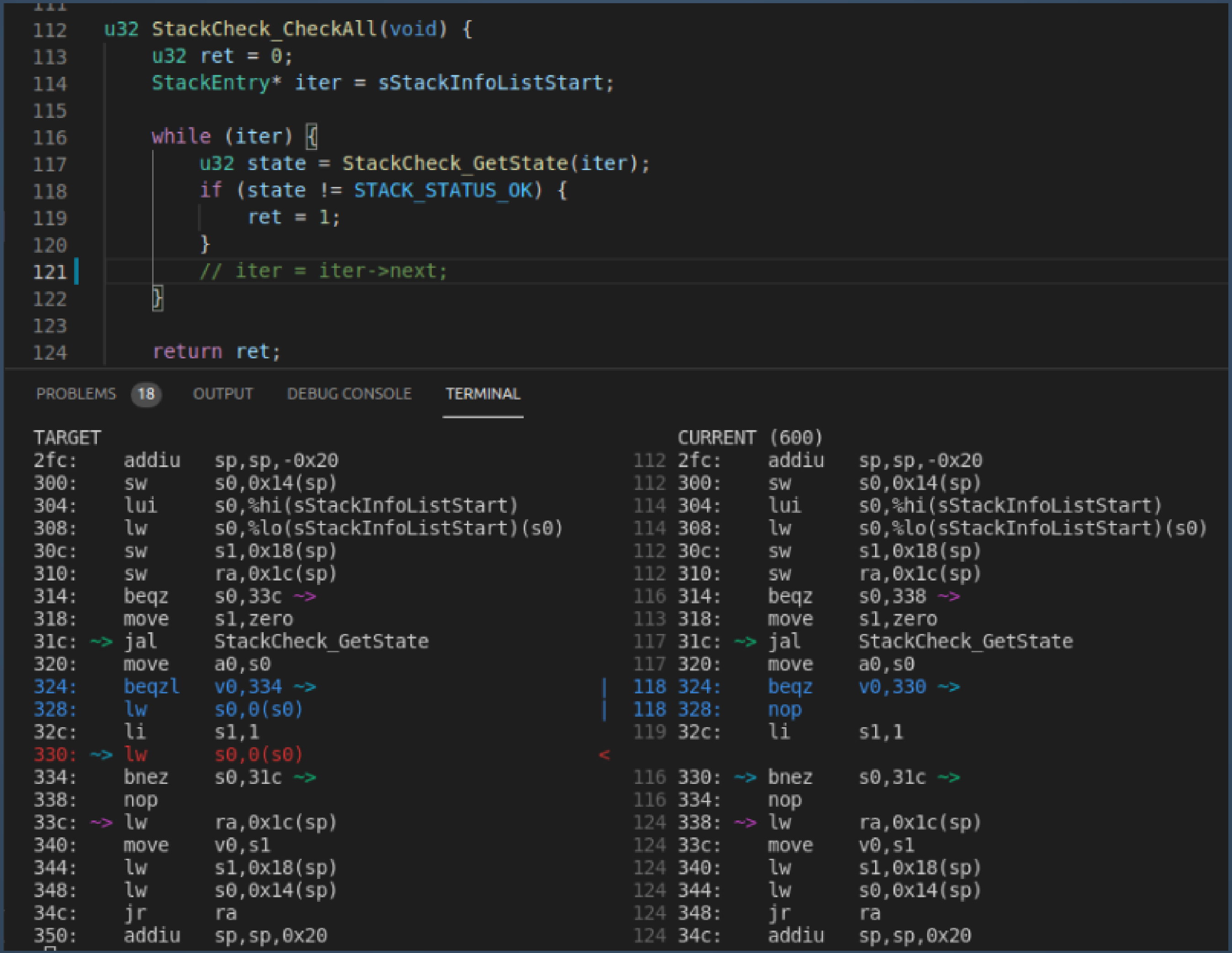
asm-differ

asm-differ

```
$ ./diff.py -mwo StackCheck_CheckAll
```

asm-differ

```
$ ./diff.py -mwo StackCheck_CheckAll
```



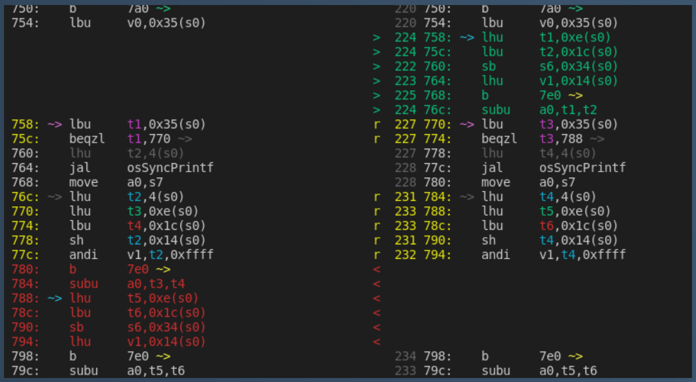
```c
u32 StackCheck_CheckAll(void) {
    u32 ret = 0;
    StackEntry* iter = sStackInfoListStart;

    while (iter) {
        u32 state = StackCheck_GetState(iter);
        if (state != STACK_STATUS_OK) {
            ret = 1;
        }
        // iter = iter->next;
    }

    return ret;
```

```
TARGET                                                   CURRENT (600)
2fc:     addiu   sp,sp,-0x20                          112 2fc:     addiu   sp,sp,-0x20
300:     sw      s0,0x14(sp)                          112 300:     sw      s0,0x14(sp)
304:     lui     s0,%hi(sStackInfoListStart)          114 304:     lui     s0,%hi(sStackInfoListStart)
308:     lw      s0,%lo(sStackInfoListStart)(s0)      114 308:     lw      s0,%lo(sStackInfoListStart)(s0)
30c:     sw      s1,0x18(sp)                          112 30c:     sw      s1,0x18(sp)
310:     sw      ra,0x1c(sp)                          112 310:     sw      ra,0x1c(sp)
314:     beqz    s0,33c ~>                            116 314:     beqz    s0,338 ~>
318:     move    s1,zero                              113 318:     move    s1,zero
31c: ~>  jal     StackCheck_GetState                  117 31c: ~>  jal     StackCheck_GetState
320:     move    a0,s0                                117 320:     move    a0,s0
324:     beqzl   v0,334 ~>                          | 118 324:     beqz    v0,330 ~>
328:     lw      s0,0(s0)                           | 118 328:     nop
32c:     li      s1,1                                 119 32c:     li      s1,1
330: ~>  lw      s0,0(s0)                           <
334:     bnez    s0,31c ~>                            116 330: ~>  bnez    s0,31c ~>
338:     nop                                          116 334:     nop
33c: ~>  lw      ra,0x1c(sp)                          124 338: ~>  lw      ra,0x1c(sp)
340:     move    v0,s1                                124 33c:     move    v0,s1
344:     lw      s1,0x18(sp)                          124 340:     lw      s1,0x18(sp)
348:     lw      s0,0x14(sp)                          124 344:     lw      s0,0x14(sp)
34c:     jr      ra                                   124 348:     jr      ra
350:     addiu   sp,sp,0x20                           124 34c:     addiu   sp,sp,0x20
```

Common problems when matching with IDO:

Common problems when matching with IDO:

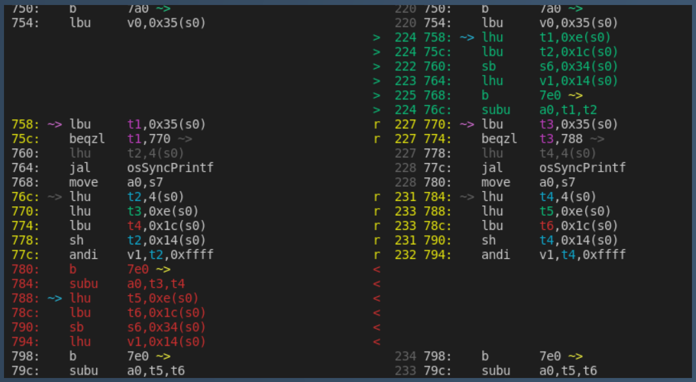
- large reordering



```
750:    b       7a0 ~>                                220 750:    b       7a0 ~>
754:    lbu     v0,0x35(s0)                           220 754:    lbu     v0,0x35(s0)
                                                    > 224 758: ~> lhu     t1,0xe(s0)
                                                    > 224 75c:    lbu     t2,0x1c(s0)
                                                    > 222 760:    sb      s6,0x34(s0)
                                                    > 223 764:    lhu     v1,0x14(s0)
                                                    > 225 768:    b       7e0 ~>
                                                    > 224 76c:    subu    a0,t1,t2
758: ~> lbu     t1,0x35(s0)                         r 227 770: ~> lbu     t3,0x35(s0)
75c:    beqzl   t1,770 ~>                           r 227 774:    beqzl   t3,788 ~>
760:    lhu     t2,4(s0)                              227 778:    lhu     t4,4(s0)
764:    jal     osSyncPrintf                          228 77c:    jal     osSyncPrintf
768:    move    a0,s7                                 228 780:    move    a0,s7
76c: ~> lhu     t2,4(s0)                            r 231 784: ~> lhu     t4,4(s0)
770:    lhu     t3,0xe(s0)                          r 233 788:    lhu     t5,0xe(s0)
774:    lbu     t4,0x1c(s0)                         r 233 78c:    lbu     t6,0x1c(s0)
778:    sh      t2,0x14(s0)                         r 231 790:    sh      t4,0x14(s0)
77c:    andi    v1,t2,0xffff                        r 232 794:    andi    v1,t4,0xffff
780:    b       7e0 ~>                              <
784:    subu    a0,t3,t4                            <
788: ~> lhu     t5,0xe(s0)                          <
78c:    lbu     t6,0x1c(s0)                         <
790:    sb      s6,0x34(s0)                         <
794:    lhu     v1,0x14(s0)                         <
798:    b       7e0 ~>                                234 798:    b       7e0 ~>
79c:    subu    a0,t5,t6                              233 79c:    subu    a0,t5,t6
```

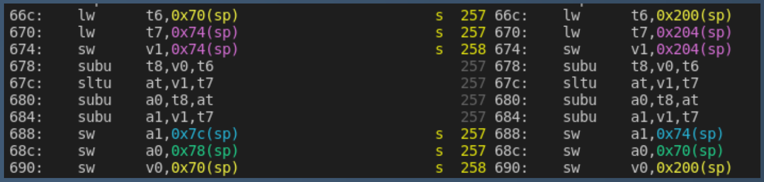
Common problems when matching with IDO:

- large reordering



```
750:     b        7a0 ~>                                220 750:     b        7a0 ~>
754:     lbu      v0,0x35(s0)                           220 754:     lbu      v0,0x35(s0)
                                                      > 224 758: ~> lhu      t1,0xe(s0)
                                                      > 224 75c:    lbu      t2,0x1c(s0)
                                                      > 222 760:    sb       s6,0x34(s0)
                                                      > 223 764:    lhu      v1,0x14(s0)
                                                      > 225 768:    b        7e0 ~>
                                                      > 224 76c:    subu     a0,t1,t2
758: ~>  lbu      t1,0x35(s0)                         r 227 770: ~> lbu      t3,0x35(s0)
75c:     beqzl    t1,770 ~>                           r 227 774:    beqzl    t3,788 ~>
760:     lhu      t2,4(s0)                              227 778:    lhu      t4,4(s0)
764:     jal      osSyncPrintf                          228 77c:    jal      osSyncPrintf
768:     move     a0,s7                                 228 780:    move     a0,s7
76c: ~>  lhu      t2,4(s0)                            r 231 784: ~> lhu      t4,4(s0)
770:     lhu      t3,0xe(s0)                          r 233 788:    lhu      t5,0xe(s0)
774:     lbu      t4,0x1c(s0)                         r 233 78c:    lbu      t6,0x1c(s0)
778:     sh       t2,0x14(s0)                         r 231 790:    sh       t4,0x14(s0)
77c:     andi     v1,t2,0xffff                        r 232 794:    andi     v1,t4,0xffff
780:     b        7e0 ~>                              <
784:     subu     a0,t3,t4                            <
788: ~>  lhu      t5,0xe(s0)                          <
78c:     lbu      t6,0x1c(s0)                         <
790:     sb       s6,0x34(s0)                         <
794:     lhu      v1,0x14(s0)                         <
798:     b        7e0 ~>                                234 798:    b        7e0 ~>
79c:     subu     a0,t5,t6                              233 79c:    subu     a0,t5,t6
```

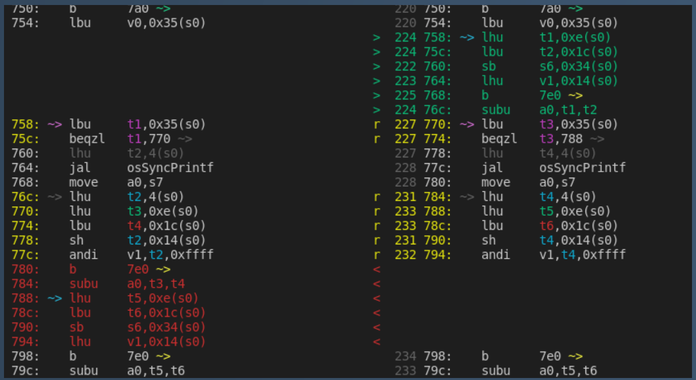
- stack placement



```
66c:     lw       t6,0x70(sp)                  s  257 66c:     lw       t6,0x200(sp)
670:     lw       t7,0x74(sp)                  s  257 670:     lw       t7,0x204(sp)
674:     sw       v1,0x74(sp)                  s  258 674:     sw       v1,0x204(sp)
678:     subu     t8,v0,t6                        257 678:     subu     t8,v0,t6
67c:     sltu     at,v1,t7                        257 67c:     sltu     at,v1,t7
680:     subu     a0,t8,at                        257 680:     subu     a0,t8,at
684:     subu     a1,v1,t7                        257 684:     subu     a1,v1,t7
688:     sw       a1,0x7c(sp)                  s  257 688:     sw       a1,0x74(sp)
68c:     sw       a0,0x78(sp)                  s  257 68c:     sw       a0,0x70(sp)
690:     sw       v0,0x70(sp)                  s  258 690:     sw       v0,0x200(sp)
```

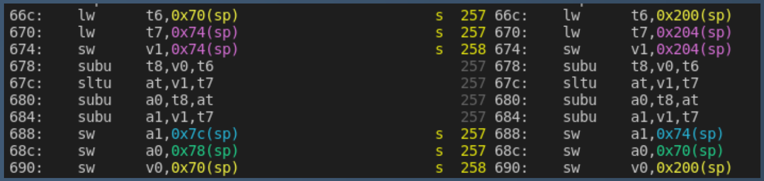
Common problems when matching with IDO:

- large reordering



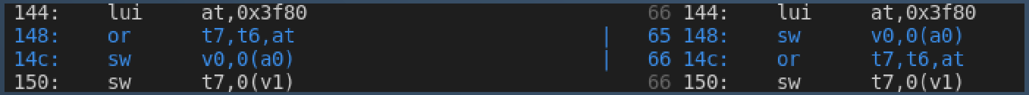
```
750:     b       7a0 ~>                                          220 750:     b       7a0 ~>
754:     lbu     v0,0x35(s0)                                     220 754:     lbu     v0,0x35(s0)
                                                             >   224 758: ~> lhu     t1,0xe(s0)
                                                             >   224 75c:    lbu     t2,0x1c(s0)
                                                             >   222 760:    sb      s6,0x34(s0)
                                                             >   223 764:    lhu     v1,0x14(s0)
                                                             >   225 768:    b       7e0 ~>
                                                             >   224 76c:    subu    a0,t1,t2
758: ~>  lbu     t1,0x35(s0)                                 r   227 770: ~> lbu     t3,0x35(s0)
75c:     beqzl   t1,770 ~>                                   r   227 774:    beqzl   t3,788 ~>
760:     lhu     t2,4(s0)                                        227 778:    lhu     t4,4(s0)
764:     jal     osSyncPrintf                                    228 77c:    jal     osSyncPrintf
768:     move    a0,s7                                           228 780:    move    a0,s7
76c: ~>  lhu     t2,4(s0)                                    r   231 784: ~> lhu     t4,4(s0)
770:     lhu     t3,0xe(s0)                                  r   233 788:    lhu     t5,0xe(s0)
774:     lbu     t4,0x1c(s0)                                 r   233 78c:    lbu     t6,0x1c(s0)
778:     sh      t2,0x14(s0)                                 r   231 790:    sh      t4,0x14(s0)
77c:     andi    v1,t2,0xffff                                r   232 794:    andi    v1,t4,0xffff
780:     b       7e0 ~>                                      <
784:     subu    a0,t3,t4                                    <
788: ~>  lhu     t5,0xe(s0)                                  <
78c:     lbu     t6,0x1c(s0)                                 <
790:     sb      s6,0x34(s0)                                 <
794:     lhu     v1,0x14(s0)                                 <
798:     b       7e0 ~>                                          234 798:    b       7e0 ~>
79c:     subu    a0,t5,t6                                        233 79c:    subu    a0,t5,t6
```

- stack placement



```
66c:    lw      t6,0x70(sp)                          s  257 66c:    lw      t6,0x200(sp)
670:    lw      t7,0x74(sp)                          s  257 670:    lw      t7,0x204(sp)
674:    sw      v1,0x74(sp)                          s  258 674:    sw      v1,0x204(sp)
678:    subu    t8,v0,t6                                257 678:    subu    t8,v0,t6
67c:    sltu    at,v1,t7                                257 67c:    sltu    at,v1,t7
680:    subu    a0,t8,at                                257 680:    subu    a0,t8,at
684:    subu    a1,v1,t7                                257 684:    subu    a1,v1,t7
688:    sw      a1,0x7c(sp)                          s  257 688:    sw      a1,0x74(sp)
68c:    sw      a0,0x78(sp)                          s  257 68c:    sw      a0,0x70(sp)
690:    sw      v0,0x70(sp)                          s  258 690:    sw      v0,0x200(sp)
```

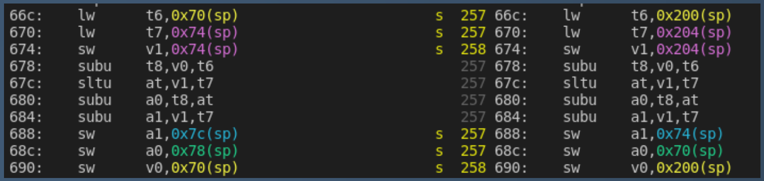
- instruction reordering



```
144:    lui     at,0x3f80                            66 144:    lui     at,0x3f80
148:    or      t7,t6,at                        |    65 148:    sw      v0,0(a0)
14c:    sw      v0,0(a0)                        |    66 14c:    or      t7,t6,at
150:    sw      t7,0(v1)                             66 150:    sw      t7,0(v1)
```

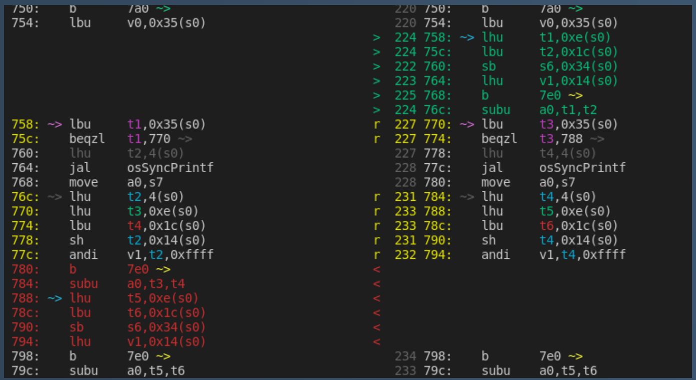
Common problems when matching with IDO:

- stack placement



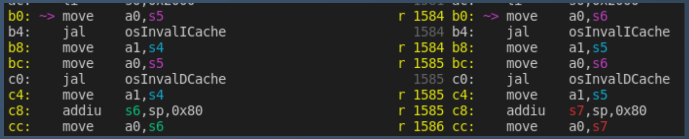
```
66c:     lw      t6,0x70(sp)                  s  257 66c:     lw      t6,0x200(sp)
670:     lw      t7,0x74(sp)                  s  257 670:     lw      t7,0x204(sp)
674:     sw      v1,0x74(sp)                  s  258 674:     sw      v1,0x204(sp)
678:     subu    t8,v0,t6                        257 678:     subu    t8,v0,t6
67c:     sltu    at,v1,t7                        257 67c:     sltu    at,v1,t7
680:     subu    a0,t8,at                        257 680:     subu    a0,t8,at
684:     subu    a1,v1,t7                        257 684:     subu    a1,v1,t7
688:     sw      a1,0x7c(sp)                  s  257 688:     sw      a1,0x74(sp)
68c:     sw      a0,0x78(sp)                  s  257 68c:     sw      a0,0x70(sp)
690:     sw      v0,0x70(sp)                  s  258 690:     sw      v0,0x200(sp)
```

- large reordering



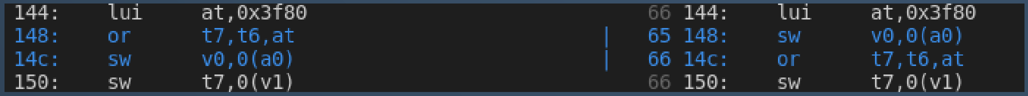
```
750:     b       7a0 ~>                          220 750:     b       7a0 ~>
754:     lbu     v0,0x35(s0)                     220 754:     lbu     v0,0x35(s0)
                                           >  224 758: ~> lhu     t1,0xe(s0)
                                           >  224 75c:    lbu     t2,0x1c(s0)
                                           >  222 760:    sb      s6,0x34(s0)
                                           >  223 764:    lhu     v1,0x14(s0)
                                           >  225 768:    b       7e0 ~>
                                           >  224 76c:    subu    a0,t1,t2
758: ~> lbu     t1,0x35(s0)                r  227 770: ~> lbu     t3,0x35(s0)
75c:    beqzl   t1,770 ~>                  r  227 774:    beqzl   t3,788 ~>
760:    lhu     t2,4(s0)                      227 778:    lhu     t4,4(s0)
764:    jal     osSyncPrintf                  228 77c:    jal     osSyncPrintf
768:    move    a0,s7                         228 780:    move    a0,s7
76c: ~> lhu     t2,4(s0)                   r  231 784: ~> lhu     t4,4(s0)
770:    lhu     t3,0xe(s0)                 r  233 788:    lhu     t5,0xe(s0)
774:    lbu     t4,0x1c(s0)                r  233 78c:    lbu     t6,0x1c(s0)
778:    sh      t2,0x14(s0)                r  231 790:    sh      t4,0x14(s0)
77c:    andi    v1,t2,0xffff               r  232 794:    andi    v1,t4,0xffff
780:    b       7e0 ~>                     <
784:    subu    a0,t3,t4                   <
788: ~> lhu     t5,0xe(s0)                 <
78c:    lbu     t6,0x1c(s0)                <
790:    sb      s6,0x34(s0)                <
794:    lhu     v1,0x14(s0)                <
798:    b       7e0 ~>                        234 798:    b       7e0 ~>
79c:    subu    a0,t5,t6                      233 79c:    subu    a0,t5,t6
```

- regalloc



```
b0: ~> move    a0,s5                       r 1584 b0: ~> move    a0,s6
b4:    jal     osInvalICache                 1584 b4:    jal     osInvalICache
b8:    move    a1,s4                       r 1584 b8:    move    a1,s5
bc:    move    a0,s5                       r 1585 bc:    move    a0,s6
c0:    jal     osInvalDCache                 1585 c0:    jal     osInvalDCache
c4:    move    a1,s4                       r 1585 c4:    move    a1,s5
c8:    addiu   s6,sp,0x80                  r 1585 c8:    addiu   s7,sp,0x80
cc:    move    a0,s6                       r 1586 cc:    move    a0,s7
```

- instruction reordering



```
144:    lui     at,0x3f80                66 144:    lui     at,0x3f80
148:    or      t7,t6,at             |   65 148:    sw      v0,0(a0)
14c:    sw      v0,0(a0)             |   66 14c:    or      t7,t6,at
150:    sw      t7,0(v1)                 66 150:    sw      t7,0(v1)
```

What can affect codegen?

What can affect codegen?

Compiler patterns

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b
- (x << 24) >> 24 ⇔ (s8)x

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b
- (x << 24) >> 24 ⇔ (s8)x
- x * 7 ⇔ (x << 2) - x

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b
- (x << 24) >> 24 ⇔ (s8)x
- x * 7 ⇔ (x << 2) - x
- loop unrolling

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b
- (x << 24) >> 24 ⇔ (s8)x
- x * 7 ⇔ (x << 2) - x
- loop unrolling
- struct copy

What can affect codegen?

Compiler patterns

- !(a ^ b) ⇔ a == b
- (x << 24) >> 24 ⇔ (s8)x
- x * 7 ⇔ (x << 2) - x
- loop unrolling
- struct copy
- deduplicating sub-expressions

What can affect codegen?

What can affect codegen?

Stupid things

What can affect codegen?

Stupid things

- if (1) { ... }

What can affect codegen?

Stupid things

- if (1) { ... }
- using temps

What can affect codegen?

Stupid things

- if (1) { ... }
- using temps
- a * -1.0 vs -a

What can affect codegen?

Stupid things

- if (1) { ... }
- using temps
- a * -1.0 vs -a
- same line / macro expansion

What can affect codegen?

Stupid things

- if (1) { ... }
- using temps
- a * -1.0 vs -a
- same line / macro expansion
- useless expressions (e.g. x & 0xFFFFFFFF)

What can affect codegen?

Stupid things

- if (1) { ... }
- using temps
- a * -1.0 vs -a
- same line / macro expansion
- useless expressions (e.g. x & 0xFFFFFFFF)

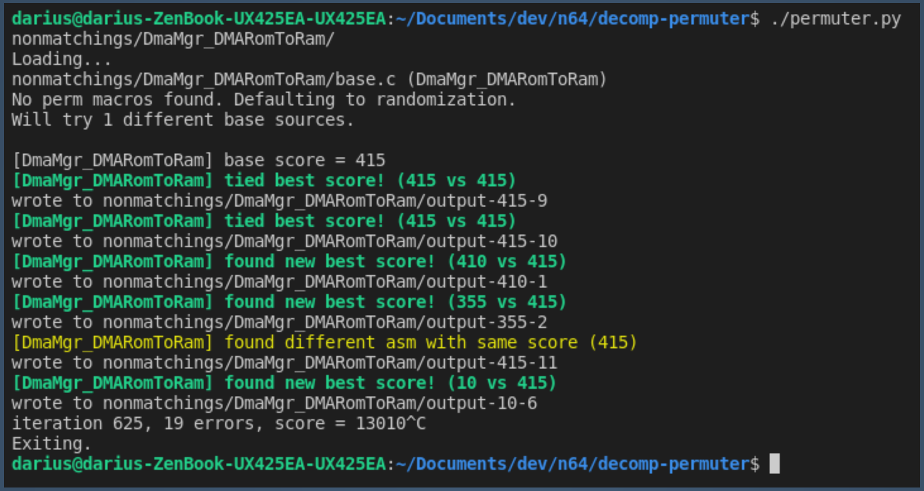
decomp-permuter

decomp-permuter

```
$ ./permuter.py nonmatchings/DmaMgr_DMARomToRam/
```

decomp-permuter

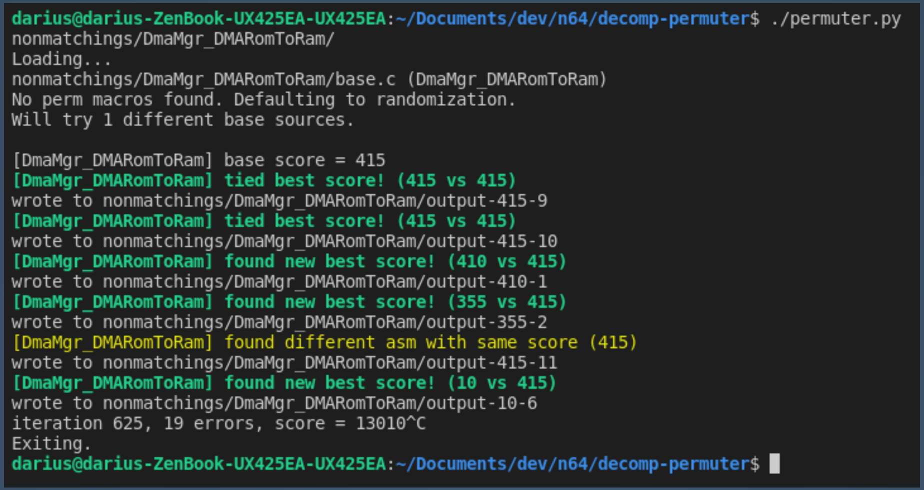
```
$ ./permuter.py nonmatchings/DmaMgr_DMARomToRam/
```



```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$ ./permuter.py
nonmatchings/DmaMgr_DMARomToRam/
Loading...
nonmatchings/DmaMgr_DMARomToRam/base.c (DmaMgr_DMARomToRam)
No perm macros found. Defaulting to randomization.
Will try 1 different base sources.

[DmaMgr_DMARomToRam] base score = 415
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-9
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-10
[DmaMgr_DMARomToRam] found new best score! (410 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-410-1
[DmaMgr_DMARomToRam] found new best score! (355 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-355-2
[DmaMgr_DMARomToRam] found different asm with same score (415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-11
[DmaMgr_DMARomToRam] found new best score! (10 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-10-6
iteration 625, 19 errors, score = 13010^C
Exiting.
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$
```

decomp-permuter

```
$ ./permuter.py nonmatchings/DmaMgr_DMARomToRam/
```

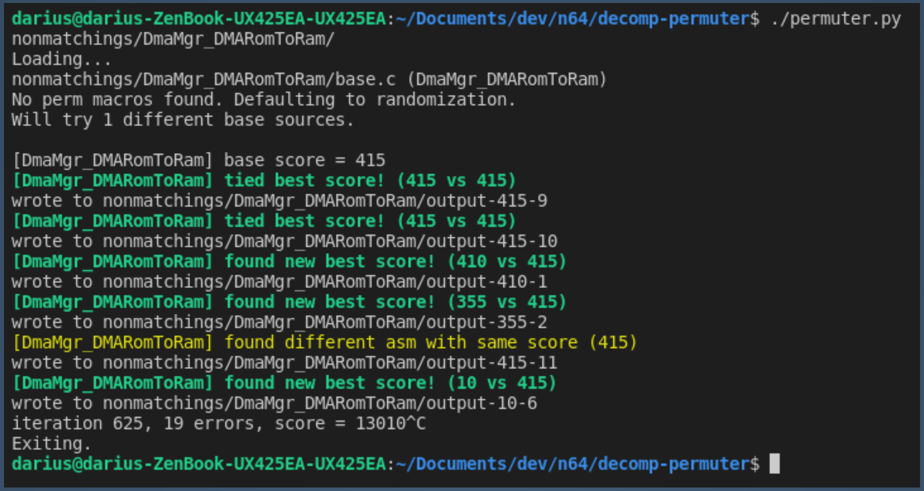


```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$ ./permuter.py
nonmatchings/DmaMgr_DMARomToRam/
Loading...
nonmatchings/DmaMgr_DMARomToRam/base.c (DmaMgr_DMARomToRam)
No perm macros found. Defaulting to randomization.
Will try 1 different base sources.

[DmaMgr_DMARomToRam] base score = 415
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-9
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-10
[DmaMgr_DMARomToRam] found new best score! (410 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-410-1
[DmaMgr_DMARomToRam] found new best score! (355 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-355-2
[DmaMgr_DMARomToRam] found different asm with same score (415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-11
[DmaMgr_DMARomToRam] found new best score! (10 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-10-6
iteration 625, 19 errors, score = 13010^C
Exiting.
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$
```

```
$ cat nonmatchings/DmaMgr_DMARomToRam/output-10-1/diff.py
```

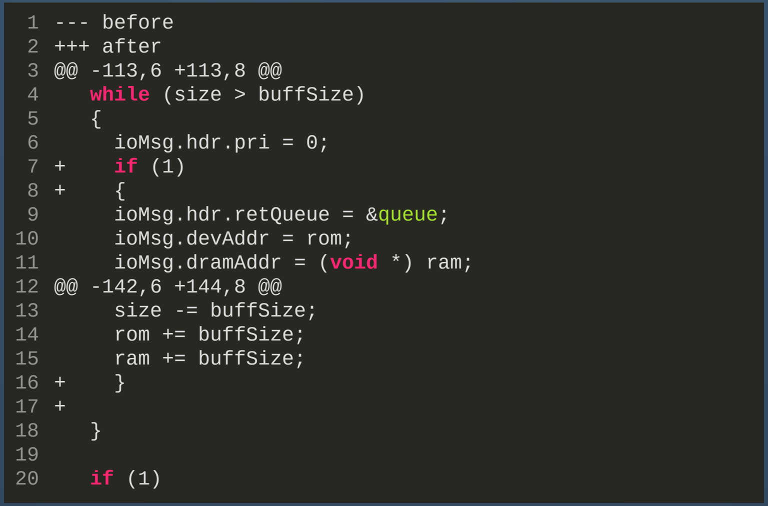
decomp-permuter

```
$ ./permuter.py nonmatchings/DmaMgr_DMARomToRam/
```



```
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$ ./permuter.py
nonmatchings/DmaMgr_DMARomToRam/
Loading...
nonmatchings/DmaMgr_DMARomToRam/base.c (DmaMgr_DMARomToRam)
No perm macros found. Defaulting to randomization.
Will try 1 different base sources.

[DmaMgr_DMARomToRam] base score = 415
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-9
[DmaMgr_DMARomToRam] tied best score! (415 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-10
[DmaMgr_DMARomToRam] found new best score! (410 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-410-1
[DmaMgr_DMARomToRam] found new best score! (355 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-355-2
[DmaMgr_DMARomToRam] found different asm with same score (415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-415-11
[DmaMgr_DMARomToRam] found new best score! (10 vs 415)
wrote to nonmatchings/DmaMgr_DMARomToRam/output-10-6
iteration 625, 19 errors, score = 13010^C
Exiting.
darius@darius-ZenBook-UX425EA-UX425EA:~/Documents/dev/n64/decomp-permuter$
```

```
$ cat nonmatchings/DmaMgr_DMARomToRam/output-10-1/diff.py
```



```
--- before
+++ after
@@ -113,6 +113,8 @@
   while (size > buffSize)
   {
     ioMsg.hdr.pri = 0;
+    if (1)
+    {
     ioMsg.hdr.retQueue = &queue;
     ioMsg.devAddr = rom;
     ioMsg.dramAddr = (void *) ram;
@@ -142,6 +144,8 @@
     size -= buffSize;
     rom += buffSize;
     ram += buffSize;
+    }
+
   }

   if (1)
```

Honorable mentions

Honorable mentions

decomp.me

Honorable mentions

decomp.me

github.com/n64decomp/ido

What about modern compilers?

What about modern compilers?

- mostly the same

What about modern compilers?

- mostly the same
- less sensitive to small differences

What about modern compilers?

- mostly the same
- less sensitive to small differences
- matching data sections might be harder

What about modern compilers?

- mostly the same
- less sensitive to small differences
- matching data sections might be harder
- better inlining

What about modern compilers?

- mostly the same
- less sensitive to small differences
- matching data sections might be harder
- better inlining
- LTO (Link Time Optimizations)

Conclusion

Links

zelda64.dev

github.com/zeldaret/oot

Questions?