МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет информационных технологий

Кафедра параллельных вычислений

ОТЧЕТ

О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ

«Разбор ассемблерного кода»

студента 2 курса, группы 20205

Муратова Максима Александровича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
Доцент
Власенко А. Ю.

Новосибирск 2021

СОДЕРЖАНИЕ

ЦЕЛИ

- Скомпилировать исходный код при разных уровнях оптимизации
- Научиться находить соответствия между ассемблерным кодом и исходным
- Сравнить ассемблерные листинги и найти реализации оптимизаций кода

ЗАДАНИЕ

1. Скомлилировать небольшую программу при уровнях оптимизации O0 и O3
2. Сопоставить участки ассемблерного кода с участками исходного кода
3. Сравнив листинги между собой, найти, какие оптимизации и как были задействованы

ОПИСАНИЕ РАБОТЫ

Для компиляции исходного кода (Приложение 1) был использована утилита `make`, с настройками компилятора g++ для архитектуры x86-64 и ключами O0 и O3. Ассемблерные листинги находятся в Приложениях 2 и 3 соответственно.

Для получения листингов достаточно ввести команду

```
$ make
```

Изменения в O3 по сравнению с O0

- В некоторых местах инструкции поменялись местами, причём такие перестановки не влияли на результат выполнения программы. Например, в процесс инициализации случайных координат попала инструкция возведения первой из них в квадрат, в то время как вторая ещё не была проинициализирована.
- Во втором ассемблерном листинге гораздо меньше обращений к оперативной памяти, вместо этого произошли обращения к регистрам настолько часто, насколько возможно.
- Длина самих используемых регистров была сокращена по возможности.
- Операция взятия остатка сократилась с 14 инструкций до 11.
- Сам ассемблерный листинг уровня оптимизации O3 содержит только 282 строки кода вместо 304 у O0.

ЗАКЛЮЧЕНИЕ

1. Для моей программы на уровне оптимизации O3 были применены следующие оптимизации: использование регистров вместо ячеек оперативной памяти настолько часто, насколько это возможно, перестановка инструкций, сокращение размеров самих регистров, использование более простых команд
2. По возможности следует избегать использование операции взятия остатка, потому что она занимает 8-10 операций (вместо 3-5 у других целочисленных операций)

Приложение 1. *main.cpp*
исходный текст программы, который был разобран на ассемблерный код

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#define PREC 10000011

using namespace std;

int main(int argc, char **argv) {
    ofstream out("output.txt");
    if(argc != 2) {
        cout << "Wrong input\n";
        out << "0\n";
        return 0;
    }
    long long dotsCount = atoll(argv[1]), goodDots = 0;
    srand(time(NULL));

    for(long long iter = 0; iter < dotsCount; iter++) {
        long long x = rand() % PREC, y = rand() % PREC;
        if(x * x + y * y <= (PREC - 1) * (PREC — 1))
            goodDots++;
    }
    out.precision(10);
    out << 4. * goodDots / dotsCount << '\n';
    out.close();
    return 0;
}
```

Приложение 2. `asm0.s`
Ассемблерный код на уровне оптимизации O0

```
    .file   "main.cpp"
    .text
    .section  .rodata
    .type   _ZStL19piecewise_construct, @object
    .size   _ZStL19piecewise_construct, 1
_ZStL19piecewise_construct:
    .zero   1
    .section  .text._ZNSt8ios_base9precisionEl,"axG",@progbits,_
ZNSt8ios_base9precisionEl,comdat
    .align 2
    .weak   _ZNSt8ios_base9precisionEl
    .type   _ZNSt8ios_base9precisionEl, @function
_ZNSt8ios_base9precisionEl:
.LFB1151:
    .cfi_startproc
    endbr64
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq %rsp, %rbp
    .cfi_def_cfa_register 6
    movq %rdi, -24(%rbp)
    movq %rsi, -32(%rbp)
    movq -24(%rbp), %rax
    movq 8(%rax), %rax
    movq %rax, -8(%rbp)
    movq -24(%rbp), %rax
    movq -32(%rbp), %rdx
    movq %rdx, 8(%rax)
    movq -8(%rbp), %rax
```

```
    popq %rbp
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE1151:
    .size    _ZNSt8ios_base9precisionEl, .-
_ZNSt8ios_base9precisionEl
    .local   _ZStL8__ioinit
    .comm    _ZStL8__ioinit,1,1
    .section  .rodata
.LC0:
    .string   "output.txt"
.LC1:
    .string   "Wrong input\n"
.LC2:
    .string   "0\n"
    .text
    .globl    main
    .type     main, @function
main:
.LFB1641:
    .cfi_startproc
    .cfi_personality 0x9b,DW.ref.__gxx_personality_v0
    .cfi_lsda 0x1b,.LLSDA1641
    endbr64
    pushq     %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq %rsp, %rbp
    .cfi_def_cfa_register 6
    pushq     %rbx
    subq $600, %rsp
```

```
        .cfi_offset 3, -24
        movl %edi, -596(%rbp)
        movq %rsi, -608(%rbp)
        movq %fs:40, %rax
        movq %rax, -24(%rbp)
        xorl %eax, %eax
        leaq -544(%rbp), %rax
        movl $16, %edx
        leaq .LC0(%rip), %rsi
        movq %rax, %rdi
    .LEHB0:
        call
_ZNSt14basic_ofstreamIcSt11char_traitsIcEEC1EPKcSt13_Ios_Openmod
e@PLT
    .LEHE0:
        cmpl $2, -596(%rbp)
        je   .L4
        leaq .LC1(%rip), %rsi
        leaq _ZSt4cout(%rip), %rdi
    .LEHB1:
        call
        _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@PLT
        leaq -544(%rbp), %rax
        leaq .LC2(%rip), %rsi
        movq %rax, %rdi
        call
        _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@PLT
        movl $0, %ebx
        jmp  .L5
    .L4:
        movq -608(%rbp), %rax
        addq $8, %rax
        movq (%rax), %rax
```

```
 movq %rax, %rdi
 call atoll@PLT
 movq %rax, -568(%rbp)
 movq $0, -584(%rbp)
 movl $0, %edi
 call time@PLT
 movl %eax, %edi
 call srand@PLT
 movq $0, -576(%rbp)
.L8:
 movq -576(%rbp), %rax
 cmpq -568(%rbp), %rax
 jge  .L6
 call rand@PLT
 movslq    %eax, %rdx
 imulq     $1125898781, %rdx, %rdx
 shrq $32, %rdx
 movl %edx, %ecx
 sarl $18, %ecx
 cltd
 subl %edx, %ecx
 movl %ecx, %edx
 imull     $1000001, %edx, %edx
 subl %edx, %eax
 movl %eax, %edx
 movslq    %edx, %rax
 movq %rax, -560(%rbp)
 call rand@PLT
 movslq    %eax, %rdx
 imulq     $1125898781, %rdx, %rdx
 shrq $32, %rdx
 movl %edx, %ecx
```

```
 sarl $18, %ecx
 cltd
 subl %edx, %ecx
 movl %ecx, %edx
 imull     $1000001, %edx, %edx
 subl %edx, %eax
 movl %eax, %edx
 movslq    %edx, %rax
 movq %rax, -552(%rbp)
 movq -560(%rbp), %rax
 imulq     %rax, %rax
 movq %rax, %rdx
 movq -552(%rbp), %rax
 imulq     %rax, %rax
 addq %rax, %rdx
 movabsq   $1000000000000, %rax
 cmpq %rax, %rdx
 jg   .L7
 addq $1, -584(%rbp)
.L7:
 addq $1, -576(%rbp)
 jmp  .L8
.L6:
 leaq -544(%rbp), %rax
 addq $248, %rax
 movl $10, %esi
 movq %rax, %rdi
 call _ZNSt8ios_base9precisionEl
 cvtsi2sdq -584(%rbp), %xmm1
 movsd     .LC3(%rip), %xmm0
 mulsd     %xmm1, %xmm0
 cvtsi2sdq -568(%rbp), %xmm1
```

```
 divsd     %xmm1, %xmm0
 leaq -544(%rbp), %rax
 movq %rax, %rdi
 call _ZNSolsEd@PLT
 movl $10, %esi
 movq %rax, %rdi
 call
_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_c@PLT
 leaq -544(%rbp), %rax
 movq %rax, %rdi
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEE5closeEv@PLT
.LEHE1:
 movl $0, %ebx
.L5:
 leaq -544(%rbp), %rax
 movq %rax, %rdi
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEED1Ev@PLT
 movl %ebx, %eax
 movq -24(%rbp), %rbx
 xorq %fs:40, %rbx
 je   .L11
 jmp  .L13
.L12:
 endbr64
 movq %rax, %rbx
 leaq -544(%rbp), %rax
 movq %rax, %rdi
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEED1Ev@PLT
 movq %rbx, %rax
 movq %rax, %rdi
.LEHB2:
 call _Unwind_Resume@PLT
```

```
.LEHE2:
.L13:
 call __stack_chk_fail@PLT
.L11:
 addq $600, %rsp
 popq %rbx
 popq %rbp
 .cfi_def_cfa 7, 8
 ret
 .cfi_endproc
.LFE1641:
 .globl   __gxx_personality_v0
 .section  .gcc_except_table,"a",@progbits
.LLSDA1641:
 .byte     0xff
 .byte     0xff
 .byte     0x1
 .uleb128 .LLSDACSE1641-.LLSDACSB1641
.LLSDACSB1641:
 .uleb128 .LEHB0-.LFB1641
 .uleb128 .LEHE0-.LEHB0
 .uleb128 0
 .uleb128 0
 .uleb128 .LEHB1-.LFB1641
 .uleb128 .LEHE1-.LEHB1
 .uleb128 .L12-.LFB1641
 .uleb128 0
 .uleb128 .LEHB2-.LFB1641
 .uleb128 .LEHE2-.LEHB2
 .uleb128 0
 .uleb128 0
.LLSDACSE1641:
```

```
    .text
    .size    main, .-main
    .type    _Z41__static_initialization_and_destruction_0ii,
@function
_Z41__static_initialization_and_destruction_0ii:
.LFB2176:
    .cfi_startproc
    endbr64
    pushq    %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq %rsp, %rbp
    .cfi_def_cfa_register 6
    subq $16, %rsp
    movl %edi, -4(%rbp)
    movl %esi, -8(%rbp)
    cmpl $1, -4(%rbp)
    jne  .L16
    cmpl $65535, -8(%rbp)
    jne  .L16
    leaq _ZStL8__ioinit(%rip), %rdi
    call _ZNSt8ios_base4InitC1Ev@PLT
    leaq __dso_handle(%rip), %rdx
    leaq _ZStL8__ioinit(%rip), %rsi
    movq _ZNSt8ios_base4InitD1Ev@GOTPCREL(%rip), %rax
    movq %rax, %rdi
    call __cxa_atexit@PLT
.L16:
    nop
    leave
    .cfi_def_cfa 7, 8
    ret
```

```
        .cfi_endproc
    .LFE2176:
        .size
        _Z41__static_initialization_and_destruction_0ii, .-
_Z41__static_initialization_and_destruction_0ii
        .type     _GLOBAL__sub_I_main, @function
    _GLOBAL__sub_I_main:
    .LFB2177:
        .cfi_startproc
        endbr64
        pushq     %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq %rsp, %rbp
        .cfi_def_cfa_register 6
        movl $65535, %esi
        movl $1, %edi
        call _Z41__static_initialization_and_destruction_0ii
        popq %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
    .LFE2177:
        .size     _GLOBAL__sub_I_main, .-_GLOBAL__sub_I_main
        .section  .init_array,"aw"
        .align 8
        .quad     _GLOBAL__sub_I_main
        .section  .rodata
        .align 8
    .LC3:
        .long     0
        .long     1074790400
        .hidden   DW.ref.__gxx_personality_v0
```

```
    .weak   DW.ref.__gxx_personality_v0
    .section  .data.rel.local.DW.ref.__gxx_personality_v0,"awG",
@progbits,DW.ref.__gxx_personality_v0,comdat
    .align 8
    .type   DW.ref.__gxx_personality_v0, @object
    .size   DW.ref.__gxx_personality_v0, 8
DW.ref.__gxx_personality_v0:
    .quad   __gxx_personality_v0
    .hidden   __dso_handle
    .ident   "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
    .section  .note.GNU-stack,"",@progbits
    .section  .note.gnu.property,"a"
    .align 8
    .long    1f - 0f
    .long    4f - 1f
    .long    5
0:
    .string   "GNU"
1:
    .align 8
    .long    0xc0000002
    .long    3f - 2f
2:
    .long    0x3
3:
    .align 8
4:
```

```
 .file     "main.cpp"
 .text
 .section  .rodata.str1.1,"aMS",@progbits,1
.LC0:
 .string   "output.txt"
.LC1:
 .string   "Wrong input\n"
.LC2:
 .string   "0\n"
 .section  .text.unlikely,"ax",@progbits
.LCOLDB4:
 .section  .text.startup,"ax",@progbits
.LHOTB4:
 .p2align 4
 .globl    main
 .type     main, @function
main:
.LFB1709:
 .cfi_startproc
 .cfi_personality 0x9b,DW.ref.__gxx_personality_v0
 .cfi_lsda 0x1b,.LLSDA1709
 endbr64
 pushq     %r15
 .cfi_def_cfa_offset 16
 .cfi_offset 15, -16
 movl $16, %edx
 pushq     %r14
 .cfi_def_cfa_offset 24
 .cfi_offset 14, -24
```

```
	pushq	%r13
	.cfi_def_cfa_offset 32
	.cfi_offset 13, -32
	pushq	%r12
	.cfi_def_cfa_offset 40
	.cfi_offset 12, -40
	pushq	%rbp
	.cfi_def_cfa_offset 48
	.cfi_offset 6, -48
	movq %rsi, %rbp
	leaq .LC0(%rip), %rsi
	pushq	%rbx
	.cfi_def_cfa_offset 56
	.cfi_offset 3, -56
	movl %edi, %ebx
	subq $552, %rsp
	.cfi_def_cfa_offset 608
	movq %fs:40, %rax
	movq %rax, 536(%rsp)
	xorl %eax, %eax
	leaq 16(%rsp), %r13
	movq %r13, %rdi
.LEHB0:
	call
_ZNSt14basic_ofstreamIcSt11char_traitsIcEEC1EPKcSt13_Ios_Openmod
e@PLT
.LEHE0:
	cmpl $2, %ebx
	je   .L2
	leaq .LC1(%rip), %rsi
	leaq _ZSt4cout(%rip), %rdi
.LEHB1:
```

```
 call
 _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@PLT
 leaq .LC2(%rip), %rsi
 movq %r13, %rdi
 call
 _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@PLT
.L6:
 movq %r13, %rdi
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEED1Ev@PLT
 movq 536(%rsp), %rax
 xorq %fs:40, %rax
 jne  .L15
 addq $552, %rsp
 .cfi_remember_state
 .cfi_def_cfa_offset 56
 xorl %eax, %eax
 popq %rbx
 .cfi_def_cfa_offset 48
 popq %rbp
 .cfi_def_cfa_offset 40
 popq %r12
 .cfi_def_cfa_offset 32
 popq %r13
 .cfi_def_cfa_offset 24
 popq %r14
 .cfi_def_cfa_offset 16
 popq %r15
 .cfi_def_cfa_offset 8
 ret
.L2:
 .cfi_restore_state
 movq 8(%rbp), %rdi
 xorl %esi, %esi
```

```
 movl $10, %edx
 call strtoll@PLT
 xorl %edi, %edi
 movq %rax, %r12
 call time@PLT
 movq %rax, %rdi
 call srand@PLT
 testq     %r12, %r12
 jle  .L9
 movabsq   $1000000000000, %r14
 xorl %r15d, %r15d
 xorl %ebp, %ebp
 .p2align 4,,10
 .p2align 3
.L5:
 call rand@PLT
 movslq    %eax, %rdx
 movq %rdx, %rbx
 imulq     $1125898781, %rdx, %rdx
 movl %ebx, %eax
 sarl $31, %eax
 sarq $50, %rdx
 subl %eax, %edx
 imull     $1000001, %edx, %edx
 subl %edx, %ebx
 call rand@PLT
 movslq    %ebx, %rbx
 movslq    %eax, %rdx
 imulq     %rbx, %rbx
 movl %eax, %ecx
 imulq     $1125898781, %rdx, %rdx
 sarl $31, %ecx
```

```
    sarq $50, %rdx
    subl %ecx, %edx
    imull     $1000001, %edx, %edx
    subl %edx, %eax
    cltq
    imulq     %rax, %rax
    addq %rbx, %rax
    cmpq %r14, %rax
    setle     %al
    addq $1, %r15
    movzbl    %al, %eax
    addq %rax, %rbp
    cmpq %r12, %r15
    jne  .L5
   .L3:
    pxor %xmm0, %xmm0
    pxor %xmm1, %xmm1
    movq %r13, %rdi
    movq $10, 272(%rsp)
    cvtsi2sdq %rbp, %xmm0
    mulsd     .LC3(%rip), %xmm0
    cvtsi2sdq %r12, %xmm1
    divsd     %xmm1, %xmm0
    call _ZNSo9_M_insertIdEERSoT_@PLT
    movq %rax, %rdi
    leaq 15(%rsp), %rsi
    movl $1, %edx
    movb $10, 15(%rsp)
    call
_ZSt16__ostream_insertIcSt11char_traitsIcEERSt13basic_ostreamIT_
T0_ES6_PKS3_l@PLT
    movq %r13, %rdi
```

```
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEE5closeEv@PLT
.LEHE1:
 jmp  .L6
.L9:
 xorl %ebp, %ebp
 jmp  .L3
.L15:
 call __stack_chk_fail@PLT
.L10:
 endbr64
 movq %rax, %rbp
 jmp  .L7
 .globl    __gxx_personality_v0
 .section  .gcc_except_table,"a",@progbits
.LLSDA1709:
 .byte     0xff
 .byte     0xff
 .byte     0x1
 .uleb128 .LLSDACSE1709-.LLSDACSB1709
.LLSDACSB1709:
 .uleb128 .LEHB0-.LFB1709
 .uleb128 .LEHE0-.LEHB0
 .uleb128 0
 .uleb128 0
 .uleb128 .LEHB1-.LFB1709
 .uleb128 .LEHE1-.LEHB1
 .uleb128 .L10-.LFB1709
 .uleb128 0
.LLSDACSE1709:
 .section  .text.startup
 .cfi_endproc
 .section  .text.unlikely
```

```
 .cfi_startproc
 .cfi_personality 0x9b,DW.ref.__gxx_personality_v0
 .cfi_lsda 0x1b,.LLSDAC1709
 .type     main.cold, @function
main.cold:
.LFSB1709:
.L7:
 .cfi_def_cfa_offset 608
 .cfi_offset 3, -56
 .cfi_offset 6, -48
 .cfi_offset 12, -40
 .cfi_offset 13, -32
 .cfi_offset 14, -24
 .cfi_offset 15, -16
 movq %r13, %rdi
 call _ZNSt14basic_ofstreamIcSt11char_traitsIcEED1Ev@PLT
 movq %rbp, %rdi
.LEHB2:
 call _Unwind_Resume@PLT
.LEHE2:
 .cfi_endproc
.LFE1709:
 .section  .gcc_except_table
.LLSDAC1709:
 .byte     0xff
 .byte     0xff
 .byte     0x1
 .uleb128 .LLSDACSEC1709-.LLSDACSBC1709
.LLSDACSBC1709:
 .uleb128 .LEHB2-.LCOLDB4
 .uleb128 .LEHE2-.LEHB2
 .uleb128 0
```

```
 .uleb128 0
.LLSDACSEC1709:
 .section  .text.unlikely
 .section  .text.startup
 .size     main, .-main
 .section  .text.unlikely
 .size     main.cold, .-main.cold
.LCOLDE4:
 .section  .text.startup
.LHOTE4:
 .p2align 4
 .type     _GLOBAL__sub_I_main, @function
_GLOBAL__sub_I_main:
.LFB2244:
 .cfi_startproc
 endbr64
 subq $8, %rsp
 .cfi_def_cfa_offset 16
 leaq _ZStL8__ioinit(%rip), %rdi
 call _ZNSt8ios_base4InitC1Ev@PLT
 movq _ZNSt8ios_base4InitD1Ev@GOTPCREL(%rip), %rdi
 addq $8, %rsp
 .cfi_def_cfa_offset 8
 leaq __dso_handle(%rip), %rdx
 leaq _ZStL8__ioinit(%rip), %rsi
 jmp  __cxa_atexit@PLT
 .cfi_endproc
.LFE2244:
 .size     _GLOBAL__sub_I_main, .-_GLOBAL__sub_I_main
 .section  .init_array,"aw"
 .align 8
 .quad     _GLOBAL__sub_I_main
```

```
	.local	_ZStL8__ioinit
	.comm	_ZStL8__ioinit,1,1
	.section	.rodata.cst8,"aM",@progbits,8
	.align 8
.LC3:
	.long	0
	.long	1074790400
	.hidden	DW.ref.__gxx_personality_v0
	.weak	DW.ref.__gxx_personality_v0
	.section	.data.rel.local.DW.ref.__gxx_personality_v0,"awG",
@progbits,DW.ref.__gxx_personality_v0,comdat
	.align 8
	.type	DW.ref.__gxx_personality_v0, @object
	.size	DW.ref.__gxx_personality_v0, 8
DW.ref.__gxx_personality_v0:
	.quad	__gxx_personality_v0
	.hidden	__dso_handle
	.ident	"GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
	.section	.note.GNU-stack,"",@progbits
	.section	.note.gnu.property,"a"
	.align 8
	.long	 1f - 0f
	.long	 4f - 1f
	.long	 5
0:
	.string	 "GNU"
1:
	.align 8
	.long	 0xc0000002
	.long	 3f - 2f
2:
	.long	 0x3
```

```
3:
 .align 8
4:
```

Приложение 4. `Makefile`
Файл получения ассемблерных листингов

```
CC=g++

all: asm0.s asm3.s

asm0.s: main.cpp
    $(CC) -O0 -S main.cpp -o asm0.s

asm3.s: main.cpp
    $(CC) -O3 -S main.cpp -o asm3.s
```

- Справочник по регистрам архитектуры x86-64
  http://www.ccfit.nsu.ru/~kireev/lab2/lab2reg.htm
- Справочники по командам ассемблерного кода архитектуры x86-64 (для Intel, но в 2T&T все команды аналогичны)
  http://www.ccfit.nsu.ru/~kireev/lab2/lab2com.htm
  https://www.felixcloutier.com/x86/index.html