Mode: All
Left file: D:\workspace\study\Code-Optimization\MIT-6.172\Lectures\Week3\MIT6_172F18_hw3\homework3\loop-nv.s
Right file: D:\workspace\study\Code-Optimization\MIT-6.172\Lectures\Week3\MIT6_172F18_hw3\homework3\loop-v.s

```
        .text                                   =       .text
        .file   "loop.c"                                .file   "loop.c"
        .section        .rodata.cst8,"aM",@p            .section        .rodata.cst8,"aM",@p
» rogbits,8                                     » rogbits,8
        .p2align        3               # --            .p2align        3               # --
»  Begin function main                        »  Begin function main
.LCPI0_0:                                       .LCPI0_0:
        .quad   4472406533629990549     # do            .quad   4472406533629990549     # do
» uble 1.0000000000000001E-9                    » uble 1.0000000000000001E-9
        .text                                           .text
        .globl  main                                    .globl  main
        .p2align        4, 0x90                         .p2align        4, 0x90
        .type   main,@function                          .type   main,@function
main:                                   # @m    main:                                   # @m
» ain                                           » ain
        .cfi_startproc                                  .cfi_startproc
# %bb.0:                                        # %bb.0:
        pushq   %r15                                    pushq   %r15
        .cfi_def_cfa_offset 16                          .cfi_def_cfa_offset 16
        pushq   %r14                                    pushq   %r14
        .cfi_def_cfa_offset 24                          .cfi_def_cfa_offset 24
        pushq   %r12                                    pushq   %r12
        .cfi_def_cfa_offset 32                          .cfi_def_cfa_offset 32
        pushq   %rbx                                    pushq   %rbx
        .cfi_def_cfa_offset 40                          .cfi_def_cfa_offset 40
        subq    $12328, %rsp            # im            subq    $12328, %rsp            # im
» m = 0x3028                                    » m = 0x3028
        .cfi_def_cfa_offset 12368                       .cfi_def_cfa_offset 12368
        .cfi_offset %rbx, -40                           .cfi_offset %rbx, -40
        .cfi_offset %r12, -32                           .cfi_offset %r12, -32
        .cfi_offset %r14, -24                           .cfi_offset %r14, -24
        .cfi_offset %r15, -16                           .cfi_offset %r15, -16
        movl    $0, 12(%rsp)                            movl    $0, 12(%rsp)
        leaq    8224(%rsp), %rdi                        leaq    8224(%rsp), %rdi
        xorl    %ebx, %ebx                              xorl    %ebx, %ebx
        xorl    %esi, %esi                              xorl    %esi, %esi
        movl    $4096, %edx             # im            movl    $4096, %edx             # im
» m = 0x1000                                    » m = 0x1000
        callq   memset                                  callq   memset
        leaq    4128(%rsp), %rdi                        leaq    4128(%rsp), %rdi
        xorl    %esi, %esi                              xorl    %esi, %esi
        movl    $4096, %edx             # im            movl    $4096, %edx             # im
» m = 0x1000                                    » m = 0x1000
        callq   memset                                  callq   memset
        leaq    32(%rsp), %rdi                          leaq    32(%rsp), %rdi
        xorl    %esi, %esi                              xorl    %esi, %esi
        movl    $4096, %edx             # im            movl    $4096, %edx             # im
» m = 0x1000                                    » m = 0x1000
        callq   memset                                  callq   memset
        leaq    16(%rsp), %rsi                          leaq    16(%rsp), %rsi
        movl    $1, %edi                                movl    $1, %edi
        callq   clock_gettime                           callq   clock_gettime
        movq    16(%rsp), %r15                          movq    16(%rsp), %r15
        movq    24(%rsp), %r14                          movq    24(%rsp), %r14
        .p2align        4, 0x90                         .p2align        4, 0x90
.LBB0_1:                                # =>    .LBB0_1:                                # =>
```

» This Loop Header: Depth=1		» This Loop Header: Depth=1
#		#
» Child Loop BB0_2 Depth 2		» Child Loop BB0_2 Depth 2
movq $-1024, %rax # im		movq $-1024, %rax # im
» m = 0xFC00		» m = 0xFC00
.p2align 4, 0x90		.p2align 4, 0x90
.LBB0_2: #		.LBB0_2: #
» Parent Loop BB0_1 Depth=1		» Parent Loop BB0_1 Depth=1
# =>		# =>
» This Inner Loop Header: Depth=2		» This Inner Loop Header: Depth=2
movl 8224(%rsp,%rax,4), %ecx	<>	movdqa 8224(%rsp,%rax,4), %xmm0
movl 8228(%rsp,%rax,4), %edx		movdqa 8240(%rsp,%rax,4), %xmm1
addl 12320(%rsp,%rax,4), %ecx		movdqa 8256(%rsp,%rax,4), %xmm2
movl %ecx, 4128(%rsp,%rax,4)		movdqa 8272(%rsp,%rax,4), %xmm3
addl 12324(%rsp,%rax,4), %edx		paddd 12320(%rsp,%rax,4), %xmm0
movl %edx, 4132(%rsp,%rax,4)		paddd 12336(%rsp,%rax,4), %xmm1
movl 8232(%rsp,%rax,4), %ecx		movdqa %xmm0, 4128(%rsp,%rax,4)
addl 12328(%rsp,%rax,4), %ecx		movdqa %xmm1, 4144(%rsp,%rax,4)
movl %ecx, 4136(%rsp,%rax,4)		paddd 12352(%rsp,%rax,4), %xmm2
movl 8236(%rsp,%rax,4), %ecx		paddd 12368(%rsp,%rax,4), %xmm3
addl 12332(%rsp,%rax,4), %ecx		movdqa %xmm2, 4160(%rsp,%rax,4)
movl %ecx, 4140(%rsp,%rax,4)		movdqa %xmm3, 4176(%rsp,%rax,4)
addq $4, %rax		addq $16, %rax
jne .LBB0_2	=	jne .LBB0_2
# %bb.3: #		# %bb.3: #
» in Loop: Header=BB0_1 Depth=1		» in Loop: Header=BB0_1 Depth=1
addl $1, %ebx		addl $1, %ebx
cmpl $100000, %ebx # im		cmpl $100000, %ebx # im
» m = 0x186A0		» m = 0x186A0
jne .LBB0_1		jne .LBB0_1
# %bb.4:		# %bb.4:
leaq 16(%rsp), %rsi		leaq 16(%rsp), %rsi
movl $1, %edi		movl $1, %edi
callq clock_gettime		callq clock_gettime
movq 16(%rsp), %r12		movq 16(%rsp), %r12
subq %r15, %r12		subq %r15, %r12
movq 24(%rsp), %rbx		movq 24(%rsp), %rbx
subq %r14, %rbx		subq %r14, %rbx
leaq 12(%rsp), %rdi		leaq 12(%rsp), %rdi
callq rand_r		callq rand_r
movl %eax, %ecx		movl %eax, %ecx
sarl $31, %ecx		sarl $31, %ecx
shrl $22, %ecx		shrl $22, %ecx
addl %eax, %ecx		addl %eax, %ecx
andl $-1024, %ecx # im		andl $-1024, %ecx # im
» m = 0xFC00		» m = 0xFC00
subl %ecx, %eax		subl %ecx, %eax
cltq		cltq
	-+	xorps %xmm1, %xmm1
cvtsi2sdq %r12, %xmm1	=	cvtsi2sdq %r12, %xmm1
	-+	xorps %xmm0, %xmm0
cvtsi2sdq %rbx, %xmm0	=	cvtsi2sdq %rbx, %xmm0
movl 32(%rsp,%rax,4), %ebx		movl 32(%rsp,%rax,4), %ebx
mulsd .LCPI0_0(%rip), %xmm0		mulsd .LCPI0_0(%rip), %xmm0
addsd %xmm1, %xmm0		addsd %xmm1, %xmm0
movl $.L.str, %edi		movl $.L.str, %edi

```
        movl    $1024, %esi             # im
» m = 0x400
        movl    $100000, %edx           # im
» m = 0x186A0
        movl    $.L.str.1, %ecx
        movl    $.L.str.2, %r8d
        movb    $1, %al
        callq   printf
        movl    %ebx, %eax
        addq    $12328, %rsp            # im
» m = 0x3028
        popq    %rbx
        popq    %r12
        popq    %r14
        popq    %r15
        retq
.Lfunc_end0:
        .size   main, .Lfunc_end0-main
        .cfi_endproc
                                        # --
»  End function
        .type   .L.str,@object          # @.
» str
        .section        .rodata.str1.1,"aMS"
» ,@progbits,1
.L.str:
        .asciz  "Elapsed execution time: %f
» sec; N: %d, I: %d, __OP__: %s, __TYPE__: %
» s\n"
        .size   .L.str, 72
        .type   .L.str.1,@object        # @.
» str.1
.L.str.1:
        .asciz  "+"
        .size   .L.str.1, 2
        .type   .L.str.2,@object        # @.
» str.2
.L.str.2:
        .asciz  "uint32_t"
        .size   .L.str.2, 9
        .ident  "clang version 6.0.0-1ubuntu
» 2 (tags/RELEASE_600/final)"
        .section        ".note.GNU-stack",""
» ,@progbits
```

```
        movl    $1024, %esi             # im
» m = 0x400
        movl    $100000, %edx           # im
» m = 0x186A0
        movl    $.L.str.1, %ecx
        movl    $.L.str.2, %r8d
        movb    $1, %al
        callq   printf
        movl    %ebx, %eax
        addq    $12328, %rsp            # im
» m = 0x3028
        popq    %rbx
        popq    %r12
        popq    %r14
        popq    %r15
        retq
.Lfunc_end0:
        .size   main, .Lfunc_end0-main
        .cfi_endproc
                                        # --
»  End function
        .type   .L.str,@object          # @.
» str
        .section        .rodata.str1.1,"aMS"
» ,@progbits,1
.L.str:
        .asciz  "Elapsed execution time: %f
» sec; N: %d, I: %d, __OP__: %s, __TYPE__: %
» s\n"
        .size   .L.str, 72
        .type   .L.str.1,@object        # @.
» str.1
.L.str.1:
        .asciz  "+"
        .size   .L.str.1, 2
        .type   .L.str.2,@object        # @.
» str.2
.L.str.2:
        .asciz  "uint32_t"
        .size   .L.str.2, 9
        .ident  "clang version 6.0.0-1ubuntu
» 2 (tags/RELEASE_600/final)"
        .section        ".note.GNU-stack",""
» ,@progbits
```