1. 어떤 파일에서 읽는가?
→ /usr/share/dict/american-english 파일에서 읽는다.

2. 총 몇 바이트를 읽는가?
→ 코드에서는 읽을 데이터 크기를 1024 바이트로 설정했다.

→ ls -l /usr/share/dict/american-english 결과를 보면 파일 크기가 985,084 바이트이다.

코드상 기본 설정은 1024 바이트, 전체 파일을 읽고 싶다면 985,084 바이트를 읽게 수정 가능하다.

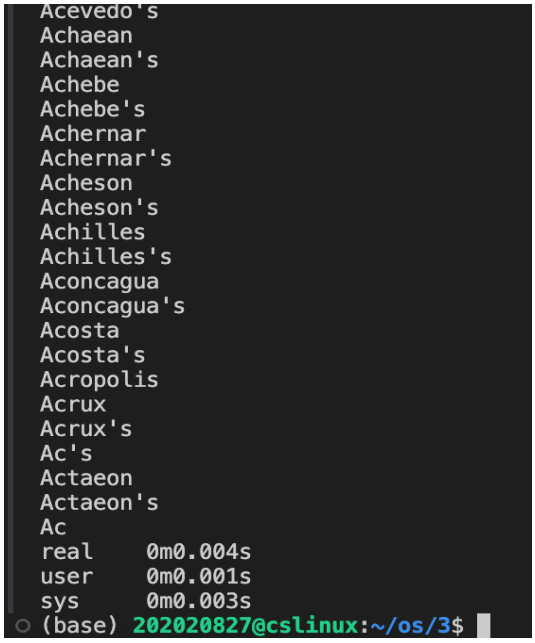
3. 사용자 C 프로그램에서 system call 을 발생시키는 함수 호출은 무엇인가?
→ open() 함수: 파일 열기 시스템 콜 호출
→ read() 함수: 파일로부터 데이터 읽기 시스템 콜 호출
→ printf() 함수: 내부적으로 write() 시스템 콜을 호출하여 출력 처리

4. 어셈블리 코드에서 read 프로시저 (system call wrapper)를 호출하는 문장은 어떤 것인가?
→ 어셈블리 코드에서는 보통 call read@PLT 형태로 나타난다.

- 5. Kernel mode 및 user mode 에서 실행되는 시간은 각각 얼마인가?

time ./lowlevel_read 사용해서 확인.

- Kernel mode (sys time): 0.003 초
- User mode (user time): 0.001 초
- 실제 실행



- 수행 시간이 너무 짧아 측정하기 어렵다면, 다음 파일의 실제 크기를 읽어들일 크기로 설정
  (ls -l /usr/share/dict/american-english 명령을 통해 파일 크기 확인)

```
(base) 202020827@cslinux:~/os/3$ ls -l /usr/share/dict/american-english
-rw-r--r-- 1 root root 985084  1월  20  2022 /usr/share/dict/american-english
(base) 202020827@cslinux:~/os/3$ gcc -O3 -S -c lowlevel_read.c
(base) 202020827@cslinux:~/os/3$ more lowlevel_read.s
```

```
        .file   "lowlevel_read.c"
        .text
        .section        .rodata.str1.8,"aMS",@progbits,1
        .align 8
.LC0:
        .string "/usr/share/dict/american-english"
        .section        .text.startup,"ax",@progbits
        .p2align 4
        .globl  main
        .type   main, @function
main:
.LFB43:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        xorl    %esi, %esi
        leaq    .LC0(%rip), %rdi
        pushq   %rbx
        .cfi_def_cfa_offset 24
        .cfi_offset 3, -24
        subq    $1048, %rsp
        .cfi_def_cfa_offset 1072
        movq    %fs:40, %rax
        movq    %rax, 1032(%rsp)
        xorl    %eax, %eax
        movq    %rsp, %rbx
        call    open@PLT
        movl    $1024, %edx
        movq    %rbx, %rsi
        movl    %eax, %edi
        call    read@PLT
        testl   %eax, %eax
        jle     .L2
        subl    $1, %eax
        leaq    1(%rsp,%rax), %rbp
        .p2align 4,,10
        .p2align 3
.L3:
        movsbl  (%rbx), %edi
        addq    $1, %rbx
        call    putchar@PLT
        cmpq    %rbp, %rbx
        jne     .L3
.L2:
        movq    1032(%rsp), %rax
        subq    %fs:40, %rax
        jne     .L8
        addq    $1048, %rsp
        .cfi_remember_state
        .cfi_def_cfa_offset 24
        xorl    %eax, %eax
        popq    %rbx
        .cfi_def_cfa_offset 16
        popq    %rbp
        .cfi_def_cfa_offset 8
        ret
.L8:
        .cfi_restore_state
        call    __stack_chk_fail@PLT
        .cfi_endproc
.LFE43:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0"
        .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:
```

- objdump -d lowlevel_read에 대한 결과

```
lowlevel_read:     file format elf64-x86-64

Disassembly of section .init:

0000000000001000 <_init>:
    1000:       f3 0f 1e fa             endbr64
    1004:       48 83 ec 08             sub    $0x8,%rsp
    1008:       48 8b 05 d9 2f 00 00    mov    0x2fd9(%rip),%rax        # 3fe8 <__gmon_start__@Base>
    100f:       48 85 c0                test   %rax,%rax
    1012:       74 02                   je     1016 <_init+0x16>
    1014:       ff d0                   call   *%rax
    1016:       48 83 c4 08             add    $0x8,%rsp
    101a:       c3                      ret

Disassembly of section .plt:

0000000000001020 <.plt>:
    1020:       ff 35 82 2f 00 00       push   0x2f82(%rip)        # 3fa8
<_GLOBAL_OFFSET_TABLE_+0x8>
    1026:       f2 ff 25 83 2f 00 00    bnd jmp *0x2f83(%rip)        # 3fb0
<_GLOBAL_OFFSET_TABLE_+0x10>
    102d:       0f 1f 00                nopl   (%rax)
    1030:       f3 0f 1e fa             endbr64
    1034:       68 00 00 00 00          push   $0x0
    1039:       f2 e9 e1 ff ff ff       bnd jmp 1020 <_init+0x20>
    103f:       90                      nop
    1040:       f3 0f 1e fa             endbr64
    1044:       68 01 00 00 00          push   $0x1
    1049:       f2 e9 d1 ff ff ff       bnd jmp 1020 <_init+0x20>
    104f:       90                      nop
    1050:       f3 0f 1e fa             endbr64
    1054:       68 02 00 00 00          push   $0x2
    1059:       f2 e9 c1 ff ff ff       bnd jmp 1020 <_init+0x20>
    105f:       90                      nop
    1060:       f3 0f 1e fa             endbr64
    1064:       68 03 00 00 00          push   $0x3
    1069:       f2 e9 b1 ff ff ff       bnd jmp 1020 <_init+0x20>
    106f:       90                      nop

Disassembly of section .plt.got:

0000000000001070 <__cxa_finalize@plt>:
    1070:       f3 0f 1e fa             endbr64
    1074:       f2 ff 25 7d 2f 00 00    bnd jmp *0x2f7d(%rip)        # 3ff8
<__cxa_finalize@GLIBC_2.2.5>
    107b:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

Disassembly of section .plt.sec:

0000000000001080 <putchar@plt>:
    1080:       f3 0f 1e fa             endbr64
```

```
    1084:       f2 ff 25 2d 2f 00 00    bnd jmp *0x2f2d(%rip)        # 3fb8 <putchar@GLIBC_2.2.5>
    108b:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

0000000000001090 <__stack_chk_fail@plt>:
    1090:       f3 0f 1e fa             endbr64
    1094:       f2 ff 25 25 2f 00 00    bnd jmp *0x2f25(%rip)        # 3fc0
<__stack_chk_fail@GLIBC_2.4>
    109b:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

00000000000010a0 <read@plt>:
    10a0:       f3 0f 1e fa             endbr64
    10a4:       f2 ff 25 1d 2f 00 00    bnd jmp *0x2f1d(%rip)        # 3fc8 <read@GLIBC_2.2.5>
    10ab:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

00000000000010b0 <open@plt>:
    10b0:       f3 0f 1e fa             endbr64
    10b4:       f2 ff 25 15 2f 00 00    bnd jmp *0x2f15(%rip)        # 3fd0 <open@GLIBC_2.2.5>
    10bb:       0f 1f 44 00 00          nopl   0x0(%rax,%rax,1)

Disassembly of section .text:

00000000000010c0 <_start>:
    10c0:       f3 0f 1e fa             endbr64
    10c4:       31 ed                   xor    %ebp,%ebp
    10c6:       49 89 d1                mov    %rdx,%r9
    10c9:       5e                      pop    %rsi
    10ca:       48 89 e2                mov    %rsp,%rdx
    10cd:       48 83 e4 f0             and    $0xfffffffffffffff0,%rsp
    10d1:       50                      push   %rax
    10d2:       54                      push   %rsp
    10d3:       45 31 c0                xor    %r8d,%r8d
    10d6:       31 c9                   xor    %ecx,%ecx
    10d8:       48 8d 3d ca 00 00 00    lea    0xca(%rip),%rdi        # 11a9 <main>
    10df:       ff 15 f3 2e 00 00       call   *0x2ef3(%rip)        # 3fd8
<__libc_start_main@GLIBC_2.34>
    10e5:       f4                      hlt
    10e6:       66 2e 0f 1f 84 00 00    cs nopw 0x0(%rax,%rax,1)
    10ed:       00 00 00

00000000000010f0 <deregister_tm_clones>:
    10f0:       48 8d 3d 19 2f 00 00    lea    0x2f19(%rip),%rdi        # 4010 <__TMC_END__>
    10f7:       48 8d 05 12 2f 00 00    lea    0x2f12(%rip),%rax        # 4010 <__TMC_END__>
    10fe:       48 39 f8                cmp    %rdi,%rax
    1101:       74 15                   je     1118 <deregister_tm_clones+0x28>
    1103:       48 8b 05 d6 2e 00 00    mov    0x2ed6(%rip),%rax        # 3fe0
<_ITM_deregisterTMCloneTable@Base>
    110a:       48 85 c0                test   %rax,%rax
    110d:       74 09                   je     1118 <deregister_tm_clones+0x28>
    110f:       ff e0                   jmp    *%rax
    1111:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)
    1118:       c3                      ret
    1119:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000001120 <register_tm_clones>:
    1120:       48 8d 3d e9 2e 00 00    lea    0x2ee9(%rip),%rdi        # 4010 <__TMC_END__>
    1127:       48 8d 35 e2 2e 00 00    lea    0x2ee2(%rip),%rsi        # 4010 <__TMC_END__>
    112e:       48 29 fe                sub    %rdi,%rsi
    1131:       48 89 f0                mov    %rsi,%rax
```

```
    1134:       48 c1 ee 3f             shr    $0x3f,%rsi
    1138:       48 c1 f8 03             sar    $0x3,%rax
    113c:       48 01 c6                add    %rax,%rsi
    113f:       48 d1 fe                sar    %rsi
    1142:       74 14                   je     1158 <register_tm_clones+0x38>
    1144:       48 8b 05 a5 2e 00 00    mov    0x2ea5(%rip),%rax        # 3ff0
<_ITM_registerTMCloneTable@Base>
    114b:       48 85 c0                test   %rax,%rax
    114e:       74 08                   je     1158 <register_tm_clones+0x38>
    1150:       ff e0                   jmp    *%rax
    1152:       66 0f 1f 44 00 00       nopw   0x0(%rax,%rax,1)
    1158:       c3                      ret
    1159:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000001160 <__do_global_dtors_aux>:
    1160:       f3 0f 1e fa             endbr64
    1164:       80 3d a5 2e 00 00 00    cmpb   $0x0,0x2ea5(%rip)        # 4010 <__TMC_END__>
    116b:       75 2b                   jne    1198 <__do_global_dtors_aux+0x38>
    116d:       55                      push   %rbp
    116e:       48 83 3d 82 2e 00 00    cmpq   $0x0,0x2e82(%rip)        # 3ff8
<__cxa_finalize@GLIBC_2.2.5>
    1175:       00
    1176:       48 89 e5                mov    %rsp,%rbp
    1179:       74 0c                   je     1187 <__do_global_dtors_aux+0x27>
    117b:       48 8b 3d 86 2e 00 00    mov    0x2e86(%rip),%rdi        # 4008 <__dso_handle>
    1182:       e8 e9 fe ff ff          call   1070 <__cxa_finalize@plt>
    1187:       e8 64 ff ff ff          call   10f0 <deregister_tm_clones>
    118c:       c6 05 7d 2e 00 00 01    movb   $0x1,0x2e7d(%rip)        # 4010 <__TMC_END__>
    1193:       5d                      pop    %rbp
    1194:       c3                      ret
    1195:       0f 1f 00                nopl   (%rax)
    1198:       c3                      ret
    1199:       0f 1f 80 00 00 00 00    nopl   0x0(%rax)

00000000000011a0 <frame_dummy>:
    11a0:       f3 0f 1e fa             endbr64
    11a4:       e9 77 ff ff ff          jmp    1120 <register_tm_clones>

00000000000011a9 <main>:
    11a9:       f3 0f 1e fa             endbr64
    11ad:       55                      push   %rbp
    11ae:       48 89 e5                mov    %rsp,%rbp
    11b1:       48 81 ec 20 04 00 00    sub    $0x420,%rsp
    11b8:       64 48 8b 04 25 28 00    mov    %fs:0x28,%rax
    11bf:       00 00
    11c1:       48 89 45 f8             mov    %rax,-0x8(%rbp)
    11c5:       31 c0                   xor    %eax,%eax
    11c7:       be 00 00 00 00          mov    $0x0,%esi
    11cc:       48 8d 05 35 0e 00 00    lea    0xe35(%rip),%rax        # 2008 <_IO_stdin_used+0x8>
    11d3:       48 89 c7                mov    %rax,%rdi
    11d6:       b8 00 00 00 00          mov    $0x0,%eax
    11db:       e8 d0 fe ff ff          call   10b0 <open@plt>
    11e0:       89 85 e4 fb ff ff       mov    %eax,-0x41c(%rbp)
    11e6:       c7 85 e8 fb ff ff 00    movl   $0x400,-0x418(%rbp)
    11ed:       04 00 00
    11f0:       8b 85 e8 fb ff ff       mov    -0x418(%rbp),%eax
    11f6:       48 63 d0                movslq %eax,%rdx
    11f9:       48 8d 8d f0 fb ff ff    lea    -0x410(%rbp),%rcx
```

```
    1200:       8b 85 e4 fb ff ff       mov    -0x41c(%rbp),%eax
    1206:       48 89 ce                mov    %rcx,%rsi
    1209:       89 c7                   mov    %eax,%edi
    120b:       e8 90 fe ff ff          call   10a0 <read@plt>
    1210:       89 85 ec fb ff ff       mov    %eax,-0x414(%rbp)
    1216:       c7 85 e0 fb ff ff 00    movl   $0x0,-0x420(%rbp)
    121d:       00 00 00
    1220:       eb 21                   jmp    1243 <main+0x9a>
    1222:       8b 85 e0 fb ff ff       mov    -0x420(%rbp),%eax
    1228:       48 98                   cltq
    122a:       0f b6 84 05 f0 fb ff    movzbl -0x410(%rbp,%rax,1),%eax
    1231:       ff
    1232:       0f be c0                movsbl %al,%eax
    1235:       89 c7                   mov    %eax,%edi
    1237:       e8 44 fe ff ff          call   1080 <putchar@plt>
    123c:       83 85 e0 fb ff ff 01    addl   $0x1,-0x420(%rbp)
    1243:       8b 85 e0 fb ff ff       mov    -0x420(%rbp),%eax
    1249:       3b 85 ec fb ff ff       cmp    -0x414(%rbp),%eax
    124f:       7c d1                   jl     1222 <main+0x79>
    1251:       b8 00 00 00 00          mov    $0x0,%eax
    1256:       48 8b 55 f8             mov    -0x8(%rbp),%rdx
    125a:       64 48 2b 14 25 28 00    sub    %fs:0x28,%rdx
    1261:       00 00
    1263:       74 05                   je     126a <main+0xc1>
    1265:       e8 26 fe ff ff          call   1090 <__stack_chk_fail@plt>
    126a:       c9                      leave
    126b:       c3                      ret

Disassembly of section .fini:

000000000000126c <_fini>:
    126c:       f3 0f 1e fa             endbr64
    1270:       48 83 ec 08             sub    $0x8,%rsp
    1274:       48 83 c4 08             add    $0x8,%rsp
    1278:       c3                      ret
```