

```

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

// #define __NR_gettimeofday 96

void use_asm_syscall(unsigned long *time_buffer) {
    unsigned long syscall_nr = 96;

    asm (    "movq %0, %%rax\n"
            "movq %1, %%rdi\n"
            "movq $0, %%rsi\n"
            "syscall"
            :
            : "m" (syscall_nr), "m" (time_buffer)
            : "rax", "rdi", "rsi");
}

void use_syscall_func(unsigned long *time_buffer) {
    unsigned long syscall_nr = 96;

    syscall(syscall_nr, time_buffer, 0);
}

long nsecs(struct timespec *start, struct timespec *end) {
    return (end->tv_sec - start->tv_sec) * 1000000000 + (end->tv_nsec - start->tv_nsec);
}

#define COUNT 10000000

int main() {
    struct timespec start, end;
    unsigned long buffer[2];

    clock_gettime(CLOCK_MONOTONIC, &start);

    for (long i = 0; i < COUNT; i++) {
        use_asm_syscall(buffer);
    }

    clock_gettime(CLOCK_MONOTONIC, &end);

    printf("asm: %ld\n", nsecs(&start, &end));

    clock_gettime(CLOCK_MONOTONIC, &start);

    for (long i = 0; i < COUNT; i++) {
        use_syscall_func(buffer);
    }

    clock_gettime(CLOCK_MONOTONIC, &end);
}

```

```
    printf("syscall func: %ld\n", nsecs(&start, &end));  
    return 0;  
}
```

执行结果：

asm: 3595739386

syscall func: 3543920282

两者没有显著差别：

运行环境：

Linux iZ2ze15yksj9218jijms7sZ 5.4.0-31-generic #35-Ubuntu SMP Thu
May 7 20:20:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

分析：

直接使用syscall函数只是省去了写asm代码的工作量，运行时并没有明显的差别。平均一次系统调用大约需要350ns，即数千个指令周期。

学号：2018K8009906031