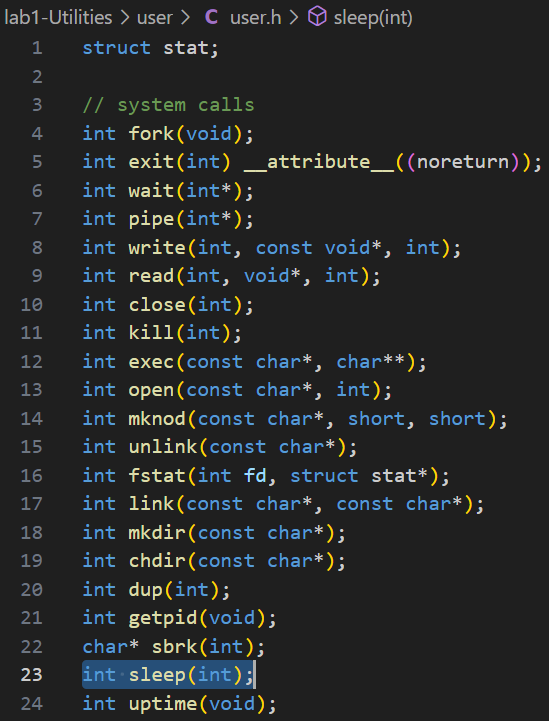
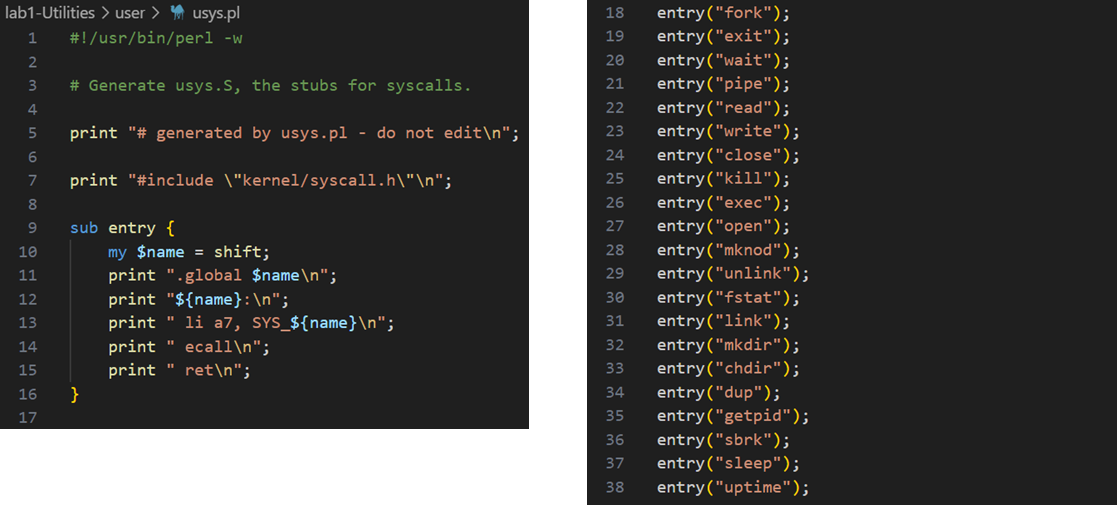
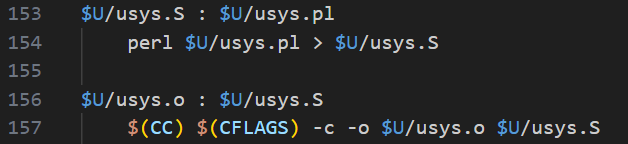
以Lab: Xv6 and Unix utilities中sleep为例解释**系统调用实现流程**

1. 用户态Sleep函数的实现

(1) sleep函数的声明

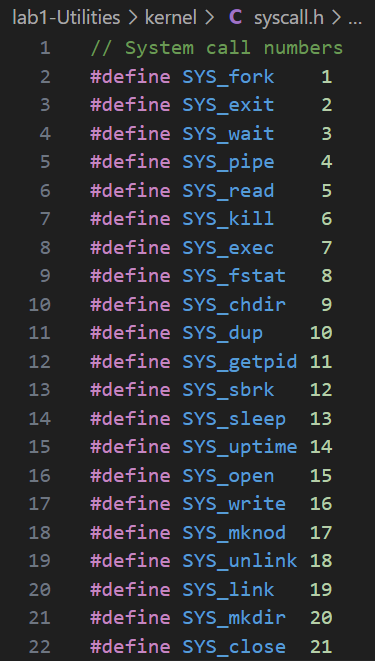
(2)sleep函数的实现

用户态Sleep函数实现较为复杂，下面一一道来。项目中有Perl文件user/usys.pl，文件内容如下：

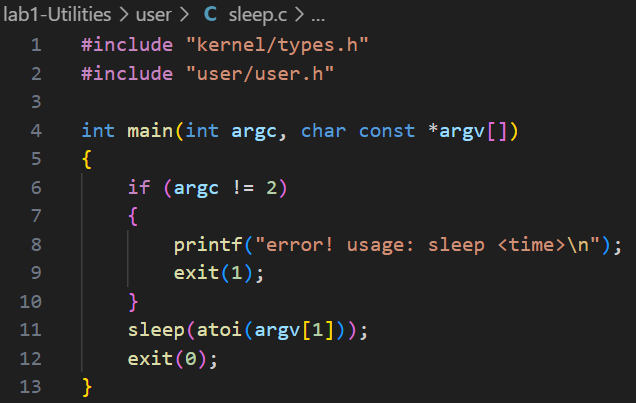
 Makefile中的命令：

我们可以看到，执行perl user/usys.pl并将输出结果重定向至user/usys.S文件中生成汇编语言，最终生成对应的可重定向文件。以entry(“sleep”)为例，汇编中.global sleep表明sleep是全局符号，可以在其他文件中调用。sleep:表明这是sleep函数的入口标签。接下来三句汇编是RISC-V指令，分别表明将SYS\_sleep加载到寄存器a7中，ecall执行系统调用，最后返回。

那么暂时我们还有一个疑问，SYS\_sleep是什么，又是在哪里定义的？我们可以看到kernel/syscall.h文件中有定义：

(3)sleep函数的调用

新建sleep.c文件，获取命令行输入，并调用sleep函数。

2. 系统调用指令ecall的实现

ecall是进行实际系统调用的入口。它是由risc-v提供的一个用于实现系统调用的指令，通常由低特权的代码发起，用来执行高特权代码，比如UserMode到SupervisorMode、SupervisorMode到MachineMode。

3. 内核态sleep函数的实现