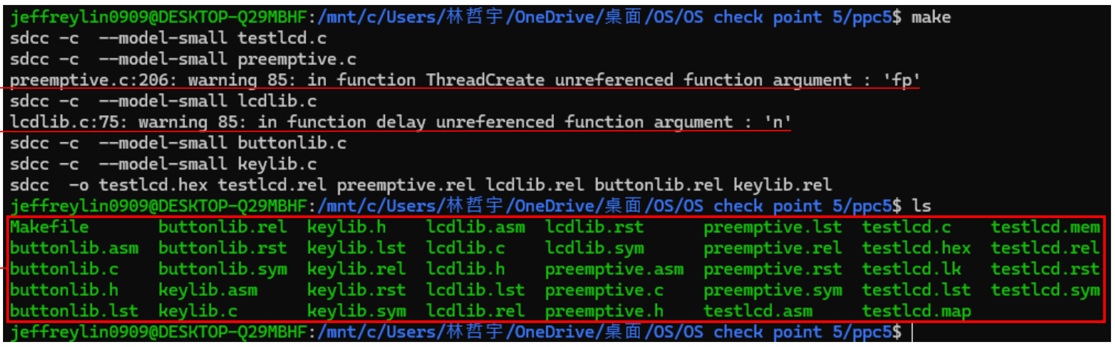
Programming Project Checkpoint 5

111062311 林哲宇

Note: I only finish ***[30 points] Peripheral devices***.

- Compilation:



```
jeffreylin0909@DESKTOP-Q29MBHF:/mnt/c/Users/林哲宇/OneDrive/桌面/OS/OS check point 5/ppc5$ make
sdcc -c  --model-small testlcd.c
sdcc -c  --model-small preemptive.c
preemptive.c:206: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc -c  --model-small lcdlib.c
lcdlib.c:75: warning 85: in function delay unreferenced function argument : 'n'
sdcc -c  --model-small buttonlib.c
sdcc -c  --model-small keylib.c
sdcc  -o testlcd.hex testlcd.rel preemptive.rel lcdlib.rel buttonlib.rel keylib.rel
jeffreylin0909@DESKTOP-Q29MBHF:/mnt/c/Users/林哲宇/OneDrive/桌面/OS/OS check point 5/ppc5$ ls
Makefile       buttonlib.rel  keylib.h    lcdlib.asm  lcdlib.rst      preemptive.lst  testlcd.c    testlcd.mem
buttonlib.asm  buttonlib.rst  keylib.lst  lcdlib.c    lcdlib.sym      preemptive.rel  testlcd.hex  testlcd.rel
buttonlib.c    buttonlib.sym  keylib.rel  lcdlib.h    preemptive.asm  preemptive.rst  testlcd.lk   testlcd.rst
buttonlib.h    keylib.asm     keylib.rst  lcdlib.lst  preemptive.c    preemptive.sym  testlcd.lst  testlcd.sym
buttonlib.lst  keylib.c       keylib.sym  lcdlib.rel  preemptive.h    testlcd.asm     testlcd.map
jeffreylin0909@DESKTOP-Q29MBHF:/mnt/c/Users/林哲宇/OneDrive/桌面/OS/OS check point 5/ppc5$
```

The warning message is because of using DPH/DPL instead of identifier of parameter, and it's totally safe.

Files generated after compilation (including .hex and .map).

- Explanation:

  I copy library from Checkpoint Description, and

  For Producer, I use library like this (button as example):

```
if (empty==0)ThreadYield();
SemaphoreWait(empty);
if (mutex==0)ThreadYield();
SemaphoreWait(mutex);
EA = 0;
if (AnyButtonPressed()){
    tmp0 = ButtonToChar();
    if (tmp0){
        buffer[b_end] = tmp0;
        b_end++;
        if (b_end==BUFFER_SIZE)b_end=0;
        EA = 1;
        SemaphoreSignal(mutex);
        SemaphoreSignal(full);
        while(AnyButtonPressed());
    }else {
        EA = 1;
        SemaphoreSignal(mutex);
        SemaphoreSignal(empty);
    }
}else {
    EA = 1;
    SemaphoreSignal(mutex);
    SemaphoreSignal(empty);
}
```

Enter semaphores (change to non-busy version for random time input of both producer)

Condition hold, produce something

Wait release

Condition does not hold, produce nothing

For Consumer, I use library like this:

```
if (full==0)ThreadYield();
SemaphoreWait(full);
if (mutex==0)ThreadYield();
SemaphoreWait(mutex);
// remove the next char from the buffer
EA = 0;
    LCD_write_char(buffer[b_start]);
    while (!LCD_ready());

    b_start++;
    if (b_start==BUFFER_SIZE)b_start=0;
EA = 1;
SemaphoreSignal(mutex);
SemaphoreSignal(empty);
```

Change to non-busy

version together

Just replace UART

transmission with LCD