```
int color;

#include "type.h"
#include "string.c"
// #include "queue.c"  // use provided queue.obj  during linking
// #include "kbd.c"    // use provided kbd.obj    during linking
#include "vid.c"
#include "exceptions.c"
#include "kernel.c"
#include "wait.c"
#include "timer.c"

void copy_vectors(void) {
    extern u32 vectors_start;
    extern u32 vectors_end;
    u32 *vectors_src = &vectors_start;
    u32 *vectors_dst = (u32 *)0;

    while(vectors_src < &vectors_end)
       *vectors_dst++ = *vectors_src++;
}
int kprintf(char *fmt, ...);

void IRQ_handler()
{
    int vicstatus, sicstatus;
    int ustatus, kstatus;

    // read VIC status register to find out which interrupt
    vicstatus = VIC_STATUS; // VIC_STATUS=0x10140000=status reg
    sicstatus = SIC_STATUS;
    if (vicstatus & 0x80000000){
       if (sicstatus & 0x08){
          kbd_handler();
       }
    }

    if (vicstatus & 0x0010){
         timer0_handler();
    }
}

int body();

int main()
{
   color = WHITE;
   row = col = 0;

   fbuf_init();
   kbd_init();

   /* enable timer0,1, uart0,1 SIC interrupts */
   VIC_INTENABLE |= (1<<4);  // timer0,1 at bit4
   VIC_INTENABLE |= (1<<5);  // timer2,3 at bit5
   VIC_INTENABLE |= (1<<31); // SIC to VIC's IRQ31

   /* enable KBD IRQ */
   SIC_ENSET = 1<<3;  // KBD int=3 on SIC
   SIC_PICENSET = 1<<3;  // KBD int=3 on SIC

   kprintf("Welcome to WANIX in Arm\n");
   init();
   kfork((int)body, 1);
```

```
   timer_init();
   timer_start(0);
   while(1){
     if (readyQueue)
        tswitch();
   }
}
```