

程序代写代做 CS 编程辅导

# COMP2300/6300

Computer Organisation and Programming



Interrupts and Asynchronism

Dr Charles Martin

Semester 1, 2022



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Interrupts

程序代写代做 CS编程辅导

- Interrupts: what, when and why
- What happens during an interrupt?
- How does it work on your microbit?
- How is this related to parallel computing?



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Admin Time

程序代写代做 CS编程辅导

You've now completed 45% of t



ment for COMP2300/6300!

midsem results, eta end of week 9.

WeChat: cstutorcs

assignment 2 coming out this week – a new challenge for you!

Assignment Project Exam Help

week 9 is an assignment pre-submission week!  
Email: [tutorcs@163.com](mailto:tutorcs@163.com)

don't forget to go to your labs!!

QQ: 749389476

<https://tutorcs.com>

## Assignment 2: Digital Pet!

程序代写代做 CS编程辅导

This is going to be fun!



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



## What is an interrupt?

程序代写代做 CS编程辅导

- Do you check your messages or notifications in times of day?
- If your phone rings, do you answer it?



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

On your microbit...

程序代写代做 CS编程辅导

- CPU is running its sequence of instructions
- How is it going to handle communications with all the other devices on the board?



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Imagine your microbit is a starship...

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Every starship has a captain

程序代写代做 CS编程辅导

(This is assumed knowledge)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

DAMAGE REPORT

A scene from Star Trek: The Next Generation. Captain Jean-Luc Picard (Patrick Stewart) is in the foreground, pointing his finger towards the viewer with a serious expression. In the background, Data (Brent Spiner) looks on with a neutral expression. The setting is the bridge of the USS Enterprise. Red text overlays are placed over the image.

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

**CAPTAIN! LOOK!**

程序代写代做 CS编程辅导



WeChat: cstutorcs

**polling vs interrupts**  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Idea 1: Polling!

程序代写代做 CS编程辅导

- Check each external device at regular intervals.
- We will need to do this regularly.



```
main: @ do some work!
bl check_keyboard
bl check_mouse
bl check_camera
bl do_some_actual_work @ yeah!
bl check_usb_1
bl check_temperature
bl do_some_other_work @ woo!
bl check_usb_2
bl check_network
b main
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Idea 2: Interrupts

程序代写代做 CS编程辅导

Change our processor architecture to add an external part to “interrupt” our fetch-decode-execute cycle!



Sequence of code is interrupted, and the CPU automatically branches to an “interrupt handler”.

WeChat: cstutorcs

```
main: @ do some work!
bl do_some_actual_work @ yeah!
bl do_some_other_work @ who
b main
```

```
keyboard_interrupt_handler:
@ do what's needed to check the keyboard
bx lr
```

Assignment Project Exam Help

Email: tutorcs@163.com

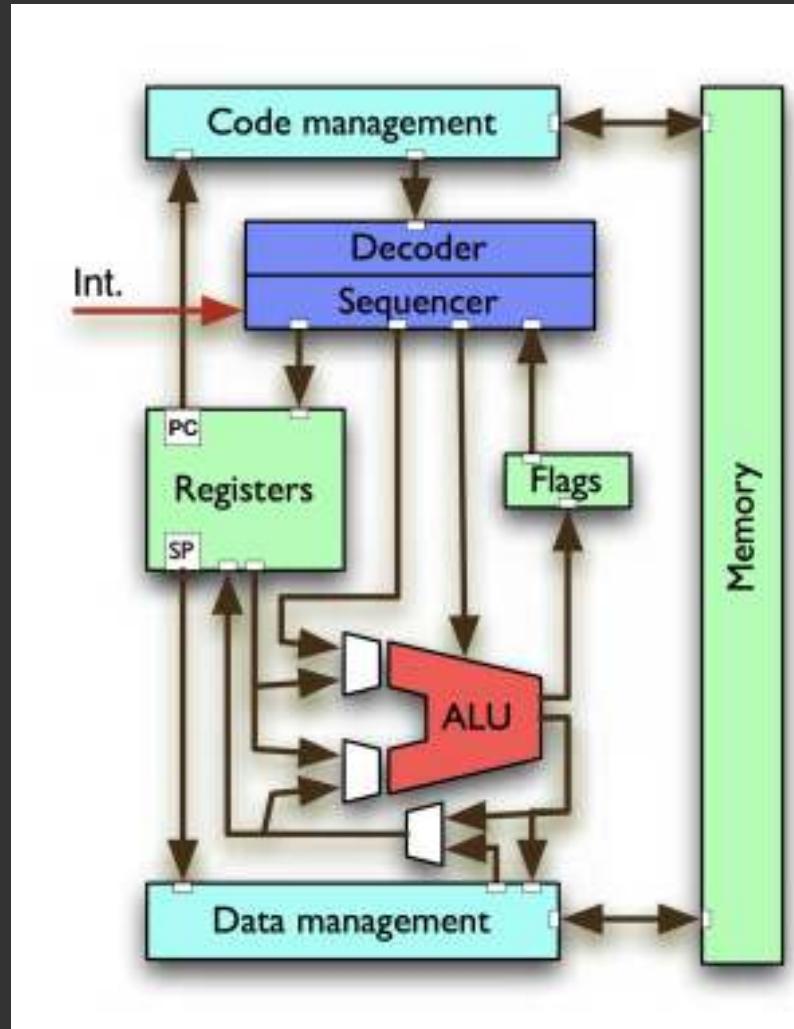
QQ: 749389476

<https://tutorcs.com>

But how does this work in practice? (we'll find out in a minute...)

# Interrupts: CPU Perspective

程序代写代做 CS 编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Polling vs Interrupts

程序代写代做 CS编程辅导

**Polling:** Simple to implement, easiest to calculate latency, fastest for small number of devices (e.g., one)



*BUT:* Devices need to wait their turn, events/data could be missed, `main` can get unmanageably long!

WeChat: cstutorcs

**Interrupts:** Enables pre-emptive scheduling, timer driven actions, transient hardware interactions, etc...

Assignment Project Exam Help

Email: tutorcs@163.com

*BUT:* A little bit more work to set up, requires external hardware (“interrupt controller”) to encode external requests.

QQ: 749389476

<https://tutorcs.com>

## Why is this important?

程序代写代做 CS编程辅导

So far, your microbit has been running in a  
clockwork (fetch-decode-execute, fetch-  
decode-execute, etc).



clockwork (fetch-decode-execute, fetch-  
decode-execute, etc).

Now, an interrupt could happen at any time.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

“An interrupt could happen at any time.”



What does this mean for your assignment as a programmer?

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Interrupts - Expecting the Unexpected



Our computers **need** to respond to all events, as well as errors!

Interrupts happen **all the time** (keyboard?)

WeChat: cstutorcs

Breaks our assumptions about fetch-decode-execute cycle, need to follow a process to save our **context**, and get back to it later.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

**How does an interrupt work?**  
Assignment Project Exam Help

---

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# How is this going to work? 程序代写代做 CS编程辅导

- How should we store the inter
- How does the microbit know h



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

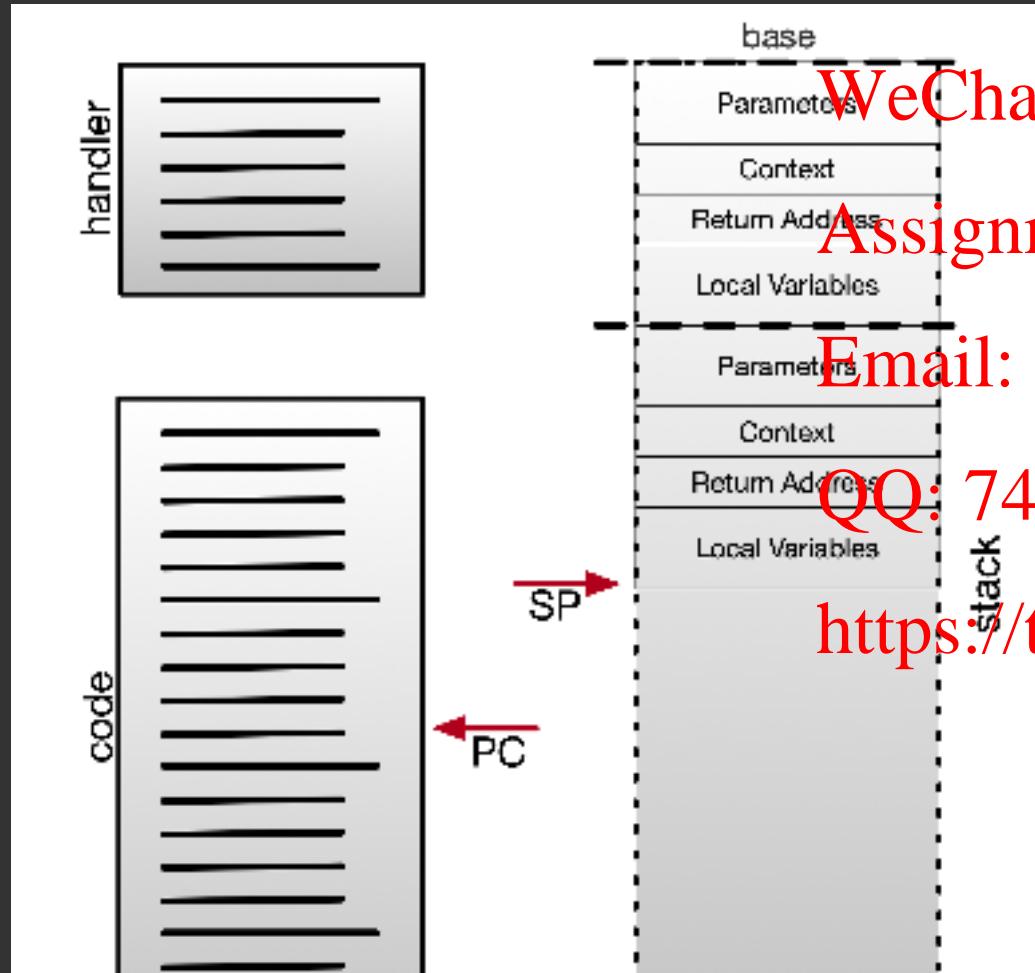
<https://tutorcs.com>

# Pre-Interrupt

程序代写代做 CS编程辅导

Everything is fine!

fetch-decode-execute ticking away



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

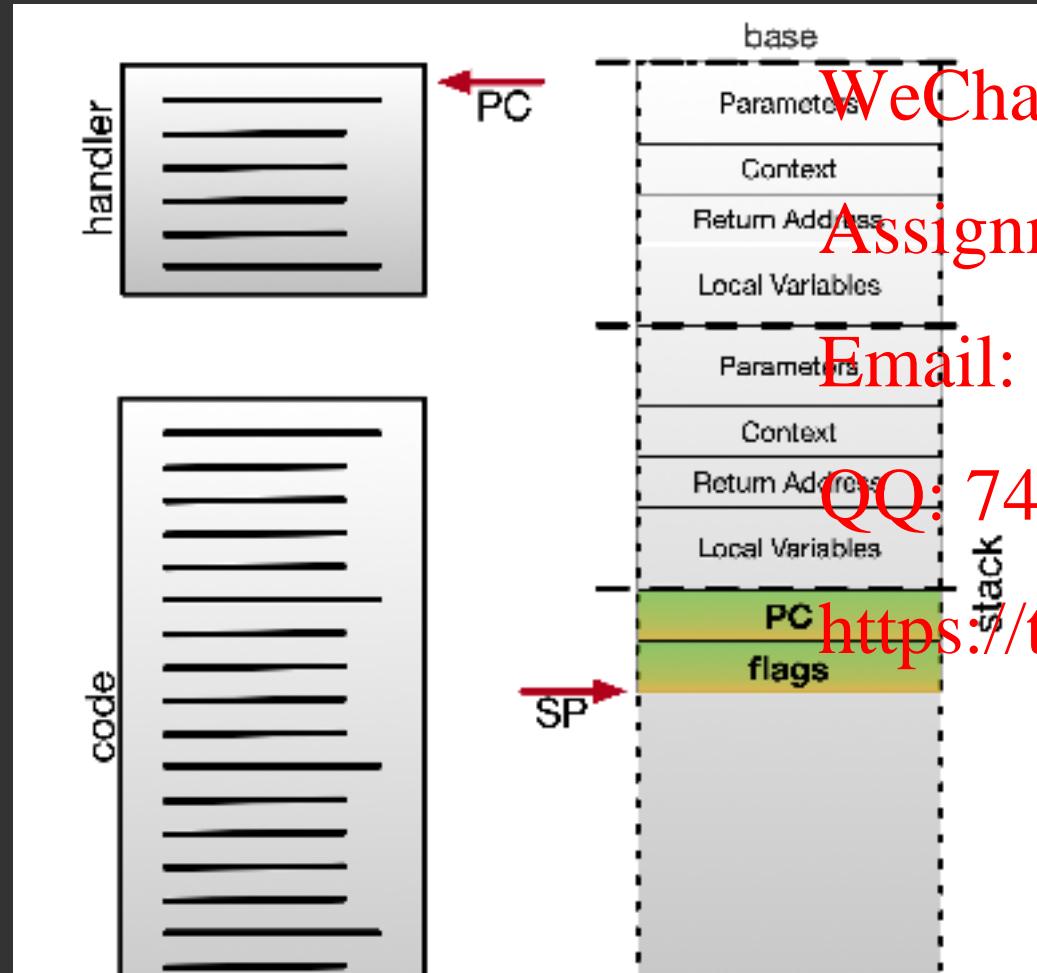
QQ: 749389476

<https://tutorcs.com>

# Interrupt occurs!!

程序代写代做 CS编程辅导

Save PC and flags on the stack (*The hardware does this for you!!*)



Now the PC is inside an **interrupt handler** (a function that has been set up previously to look after this interrupt)

WeChat: cstutorcs

Email: tutorcs@163.com  
QQ: 749389476  
Assignment Project Exam Help

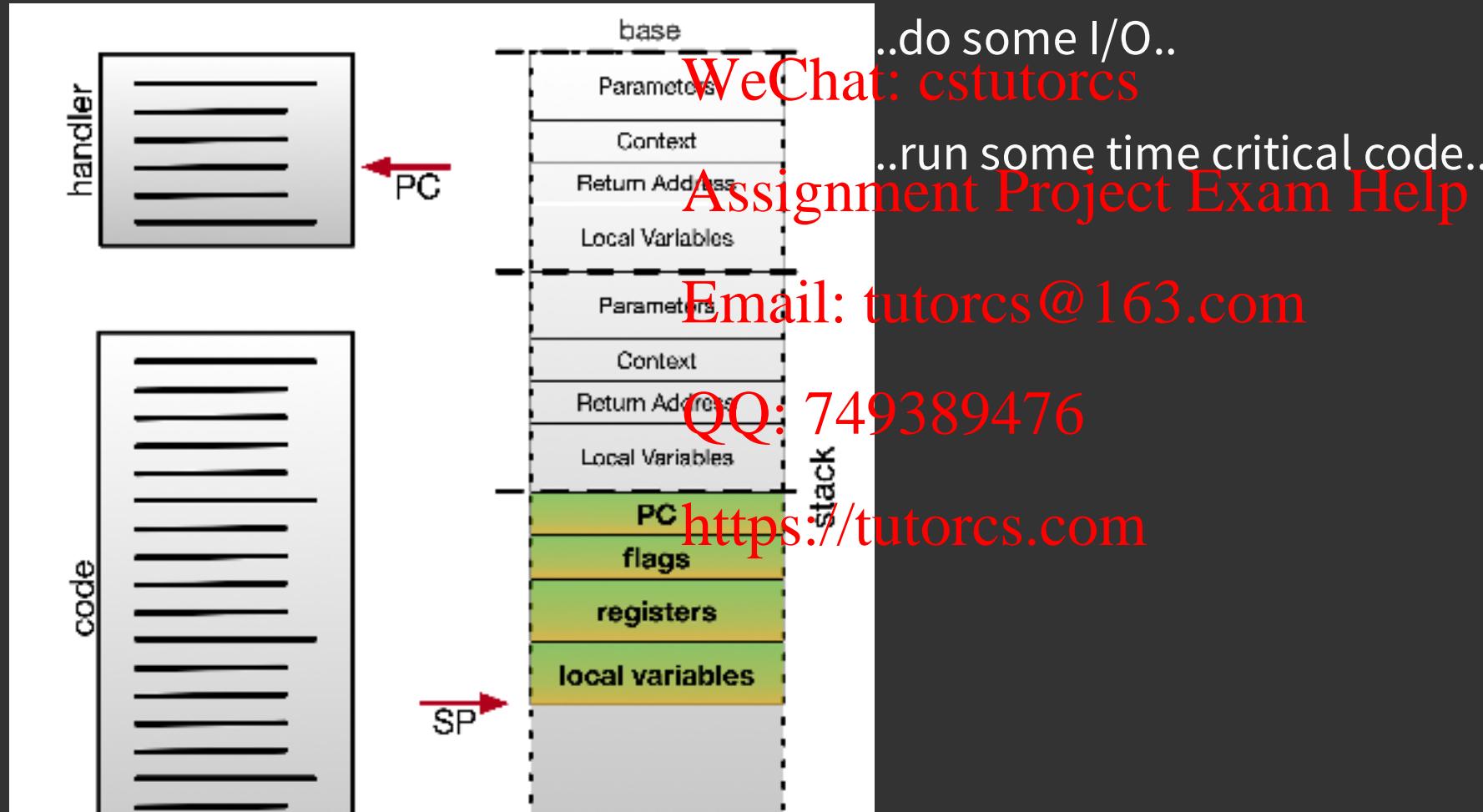
https://tutorcs.com

Run the handler code

程序代写代做 CS编程辅导

push registers

declare local variables

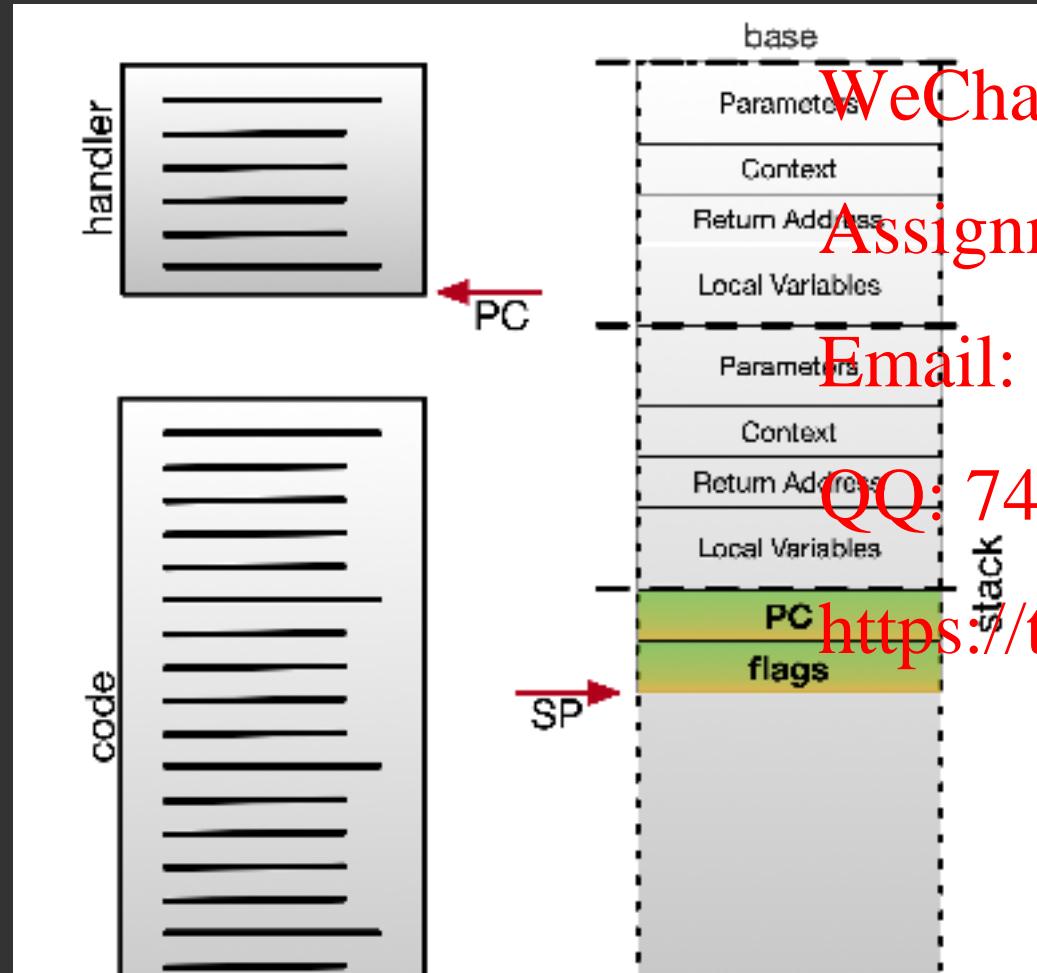


# Clean up after the handler 程序代写代做 CS编程辅导

remove local variables



pop registers



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

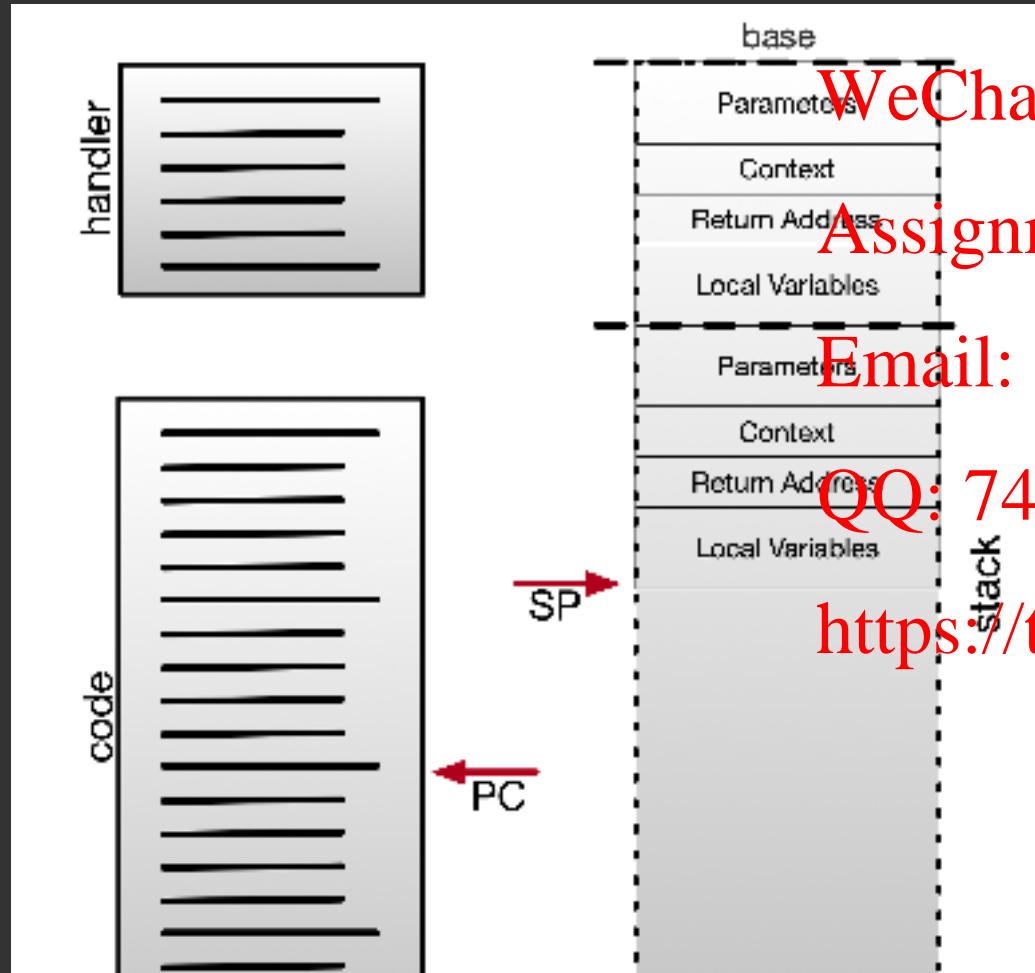
# Return from Interrupt

程序代写代做 CS编程辅导

Return from interrupt (`bx lr`)



The “special” `lr` gets the CPU to do everything back the way it was! (Clever!)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Interrupt Timeline...

程序代写代做 CS编程辅导

- Save PC and flags on the stack (*the CPU does this for you!!*)
- Now the PC is inside an **interrupt handler** (a function that has been set up previously to look after this interrupt)
- The link register **lr** has been set up with the return value!
- Do our normal function things:
  - push registers
  - declare local variables
  - ..do some I/O..
  - ..run some time critical code..
  - remove local variables
  - pop registers
- **Return from interrupt** (**bx lr**)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Things to consider

程序代写代做 CS编程辅导

- Interrupt handler code can also be interrupted!
- Which interrupts are more important? (We have **priorities** to handle this!)
- Can you overrun a stack with interrupt handlers?
- Can we have a “do not disturb” sign?



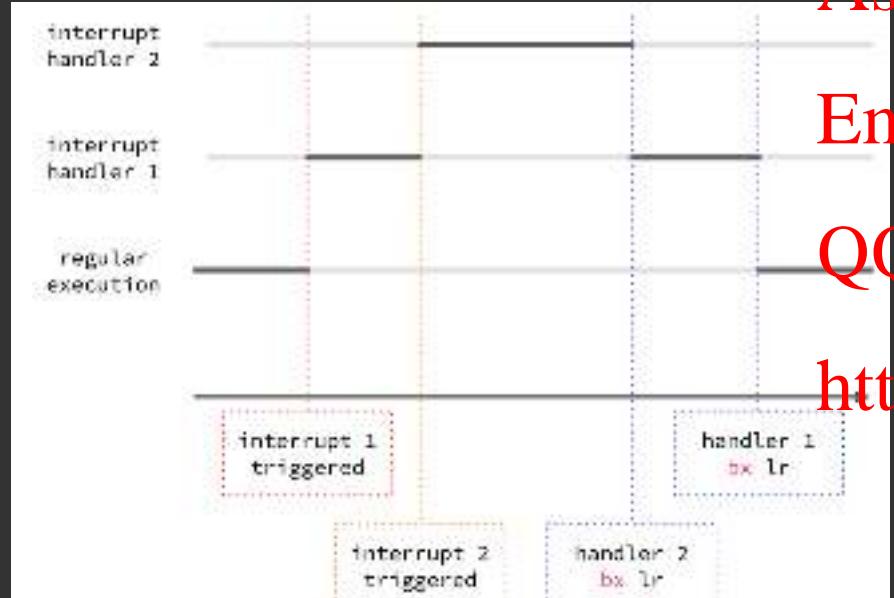
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# Interrupt Handler is a Function

程序代写代做 CS编程辅导

You get to write it! (more on that)



But you don't get to choose *when* it runs!

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

**Interrupts in Practice** Assignment Project Exam Help

---

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



You've probably experienced an interrupt  
WeChat: cstutorcs  
(exception) already!  
Assignment Project Exam Help

Email: tutorcs@163.com

What about “Usage Fault”?  
QQ: 749389476

<https://tutorcs.com>

# Guaranteed interrupt in two clock cycles



```
ldr r0, =0x20000000  
bx r0
```

N.B., the above code is broken on purpose! Your microbit will end up in the  
“Default\_Handler”

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# How do we configure an interrupt? 程序代写代做 CS编程辅导

1. Need to enable the interrupt.
2. Need to define the handler fun



3. Need to configure hardware (if using MCU features).

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Where are interrupt vectors defined? 程序代写代做 CS编程辅导

Have a look in `startup.S`:



```
.section .rodata.vtable @ (r)ead-(o)nly(data), the .vtable suffi  
.word _stack_end  
.word Reset_Handler  
.word NonMaskableInt_Handler  
.word HardFault_Handler  
.word MemManage_Handler  
...  
WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476
```

All the interrupt vectors are named <https://tutorcs.com> a default handler.

# How do we take over an interrupt handler? 程序代写代做CS编程辅导

Need to redefine one of the han



ds. E.g., for `GPIOOTE_IRQHandler`:

```
.global GPIOOTE_IRQHandler  
.type GPIOOTE_IRQHandler,%function  
GPIOOTE_IRQHandler:  
    @ do some important stuff!  
    nop  
    bx lr  
.size GPIOOTE_IRQHandler
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



So I want to use the buttons...

WeChat: cstutorcs  
Assignment Project Exam Help  
Isn't that similar to activating the LEDs?  
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

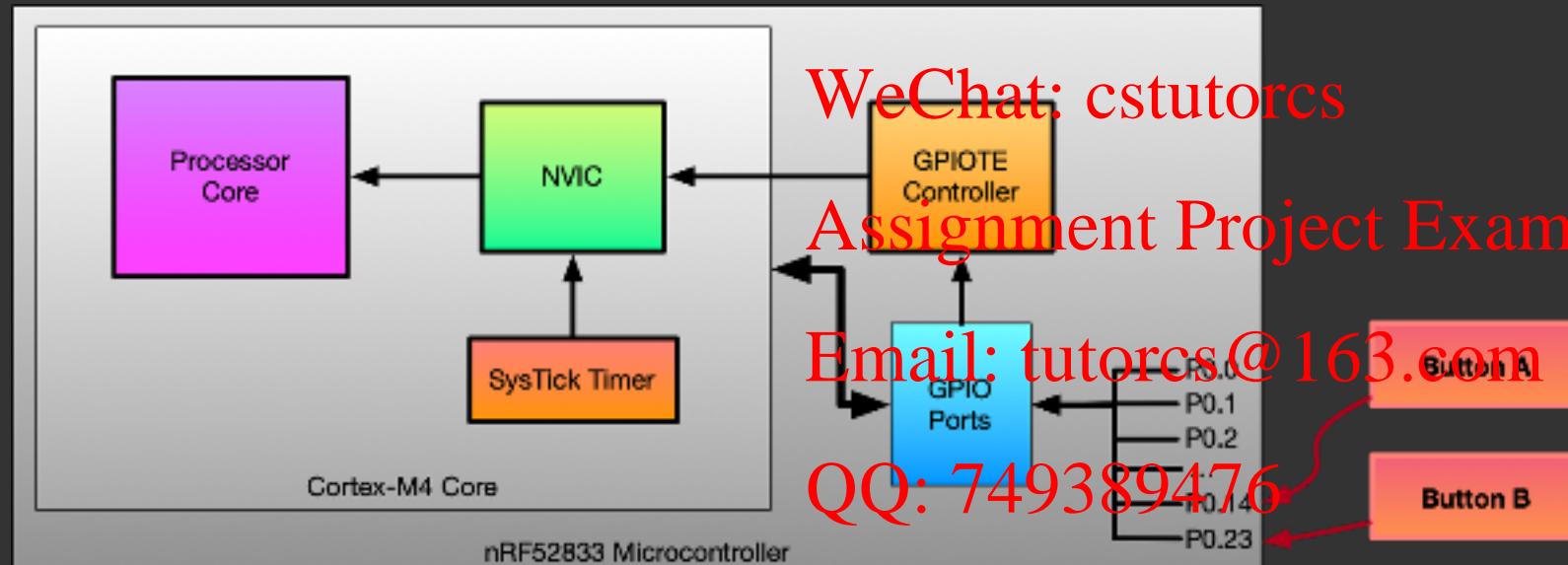
A map to the buttons...

程序代写代做 CS编程辅导

The microbit's buttons are connected to pins P0.14 (A) and P0.23 (B).



P0.14 (A) and P0.23 (B).



<https://tutorcs.com>

## Connecting it all up

程序代写代做 CS编程辅导

We need to configure the **GPIOE** to listen on **P0.14**, and configure the NVIC to make this happen. This is a little



1. Configure the GPIOE channel 0 to be in “event” mode, associate with GPIO pin 14 and port 0 (the pin and port of the A button), and to listen for low-to-high voltage events.
2. Enable interrupts for events from **Assignment Project Exam Help**
3. Enable the GPIOE interrupt in the NVIC, its ID number is “6”. For this you need to set bit “6” in the NVIC’s Interrupt Set Enable Register.
4. Implement a function called **GPIOE\_IRQHandler** and make sure you have the .global **GPIOE\_IRQHandler** directive.

<https://tutorcs.com>

Let's do it.

# Button A Interrupt Config 程序代写代做 CS编程辅导

```
@ 1: Configure GPIOE_CONFIG[0] (Section 6.9.4.1 in nRF52833 reference manual)
@ mode = 1 (event), pin = 14 and port = 4 (Pin 14 = Button A), polarity = 1 (LoToHi)
ldr r0, =GPIOE_CONFIG0
ldr r1, =(1 | 14 << 8 | 0 << 13 | 1
str r1, [r0]
```



```
@ 2: Enable Interrupt for GPIOE[0] (id = 6) (Section 6.9.4.6 in nRF52833 reference manual)
ldr r0, =GPIOE_INTENSET
ldr r1, =0b1
str r1, [r0]
```

```
@ 3: enable GPIOE (interrupt #6 = NVIC_GPIOE_ID) in NVIC_ISER0 (B3.4.4 in ARMv7-M Reference Manual)
ldr r0, =NVIC_ISER0
ldr r1, =(1 << 6) @ set the 6th bit since NVIC_GPIOE_ID=1
str r1, [r0]
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Button A Interrupt Handler



```
.global GPIOTE_IRQHandler  
.type GPIOTE_IRQHandler = function  
GPIOTE_IRQHandler:
```

@ interrupt code goes here

@ clear event

```
ldr r0, =GPIOTE_EVENTS_IN0  
ldr r1, =0  
str r1, [r0]  
bx lr
```

```
.size GPIOTE_IRQHandler, . - GPIOTE_IRQHandler
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Interrupt archaeology

程序代写代做 CS编程辅导

What about the interrupt handle

link register lr ?

status register xPSR ?

caller-save registers ( r0 - r3 )? Email: tutorcs@163.com

let's look at an interrupt handler and do some digging...



and the AAPCS?

WeChat: cstutorcs

Assignment Project Exam Help

QQ: 749389476

<https://tutorcs.com>

## Further Reading

程序代写代做 CS 编程辅导

<http://bob.cs.sonoma.edu/Intro-RPi/chp-except.html>



<http://6502.org/tutorials/interrupts.html>

WeChat: cstutorcs

<https://youtu.be/uFBNf7F3l60>

Assignment Project Exam Help

[https://learning.oreilly.com/library/view/tutorials-of-computer/9781284123043/xhtml/09\\_Chapter04\\_06.xhtml](https://learning.oreilly.com/library/view/tutorials-of-computer/9781284123043/xhtml/09_Chapter04_06.xhtml)

Email: tutorcs@163.com  
QQ: 749389476

[https://learning.oreilly.com/library/view/essentials-of-computer/9781284123043/xhtml/12\\_Chapter07\\_03.xhtml](https://learning.oreilly.com/library/view/essentials-of-computer/9781284123043/xhtml/12_Chapter07_03.xhtml)

<https://tutorcs.com>

## Questions

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

## Week 8 (part 2): Asynchronism

---

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Where were we?

程序代写代做 CS编程辅导

We just learned that an **interrupt** can stop our program *at any time* and run different code in the handler.



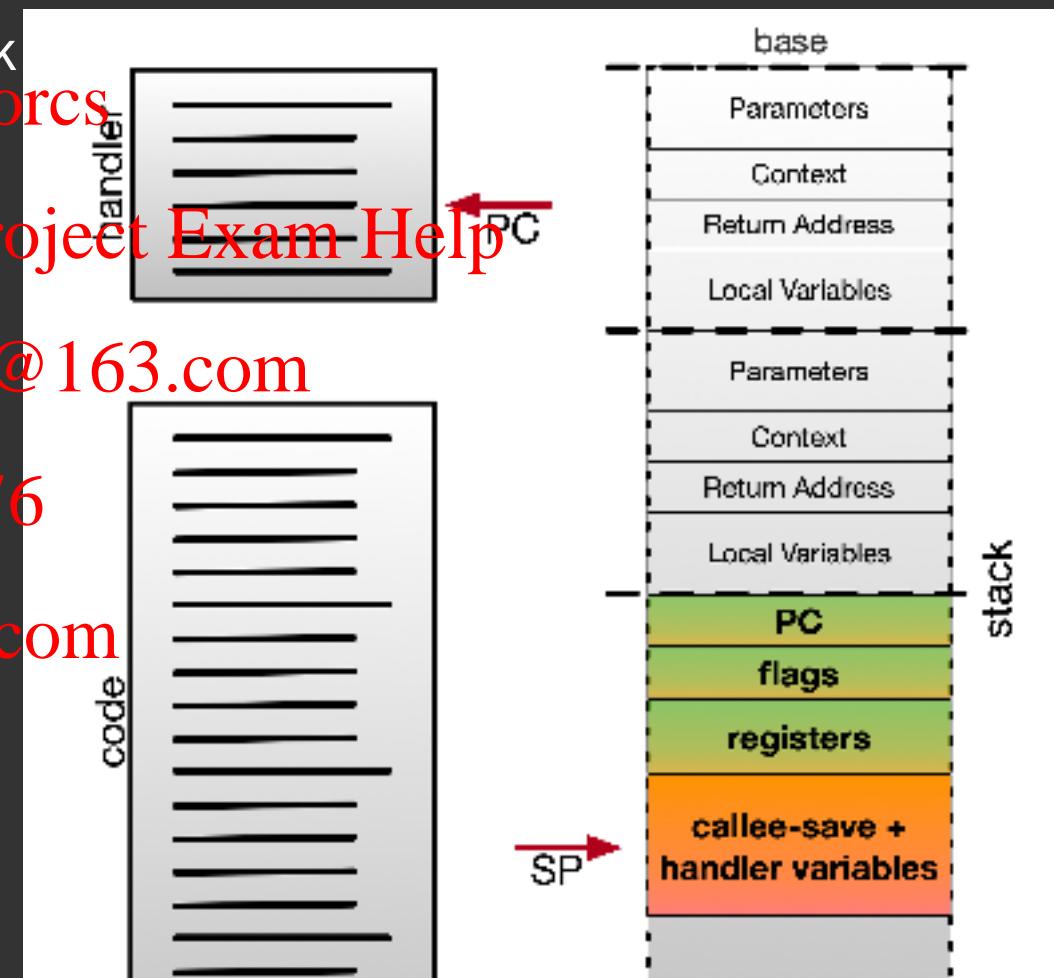
Returning from the handler puts our program back  
*just the way it was before the interrupt.*

WeChat: cstutorcs  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



SO your microbit is a starship...

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Every starship has a captain (program)

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A scene from Star Trek: The Next Generation. Captain Jean-Luc Picard (Patrick Stewart) is in the foreground, pointing his finger towards the viewer with a serious expression. In the background, Data (Brent Spiner) looks on with a neutral expression. The setting is the bridge of the USS Enterprise. Red text overlays are placed over the image.

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

**CAPTAIN! LOOK!**

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Wait a minute... can we now run two programs?

程序代写代做 CS 编程辅导

we kind of can, folks!



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Context Switching

程序代写代做 CS编程辅导

Big Idea:

Have two (or more) *main programs* and let the OS handle interrupts to swap in between them.



But what is a context?

The set of information that needs to be saved for a program to start up again after an interruption.

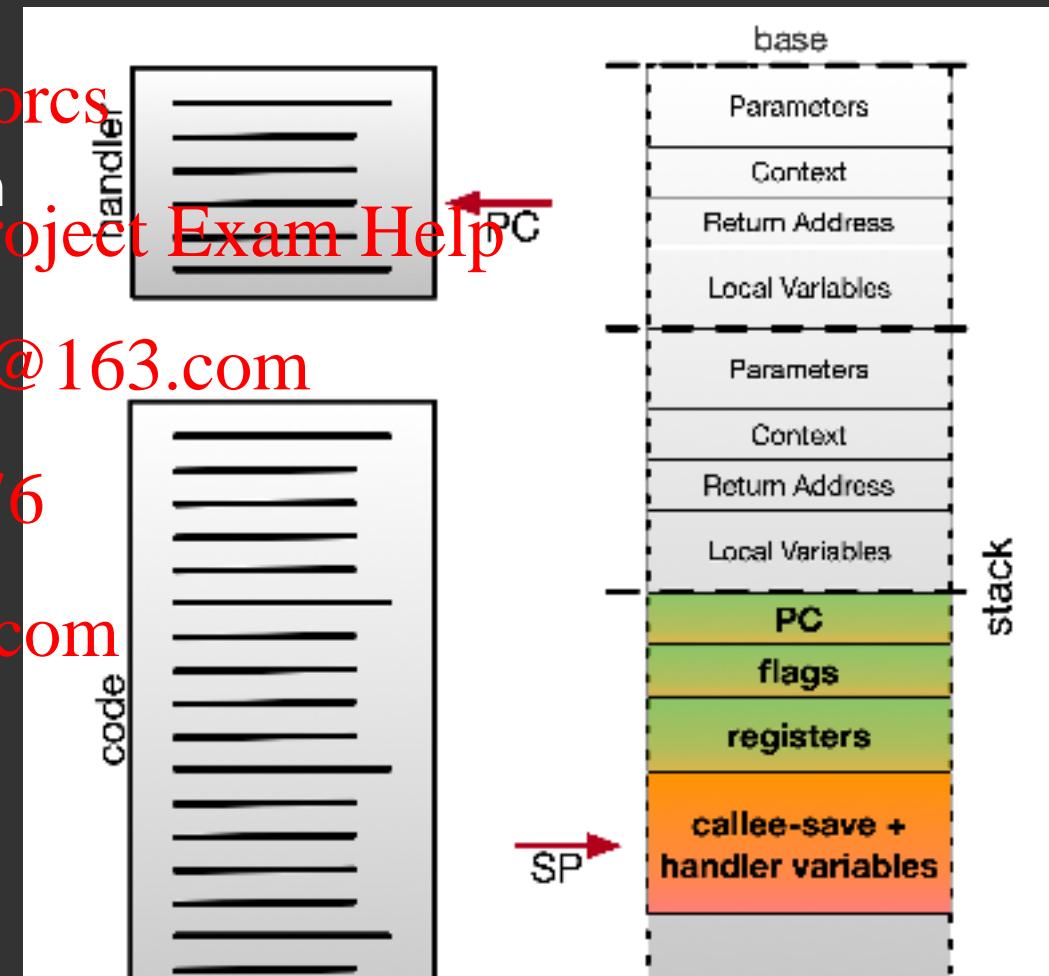
WeChat: cstutorcs

Assignment Project Exam Help

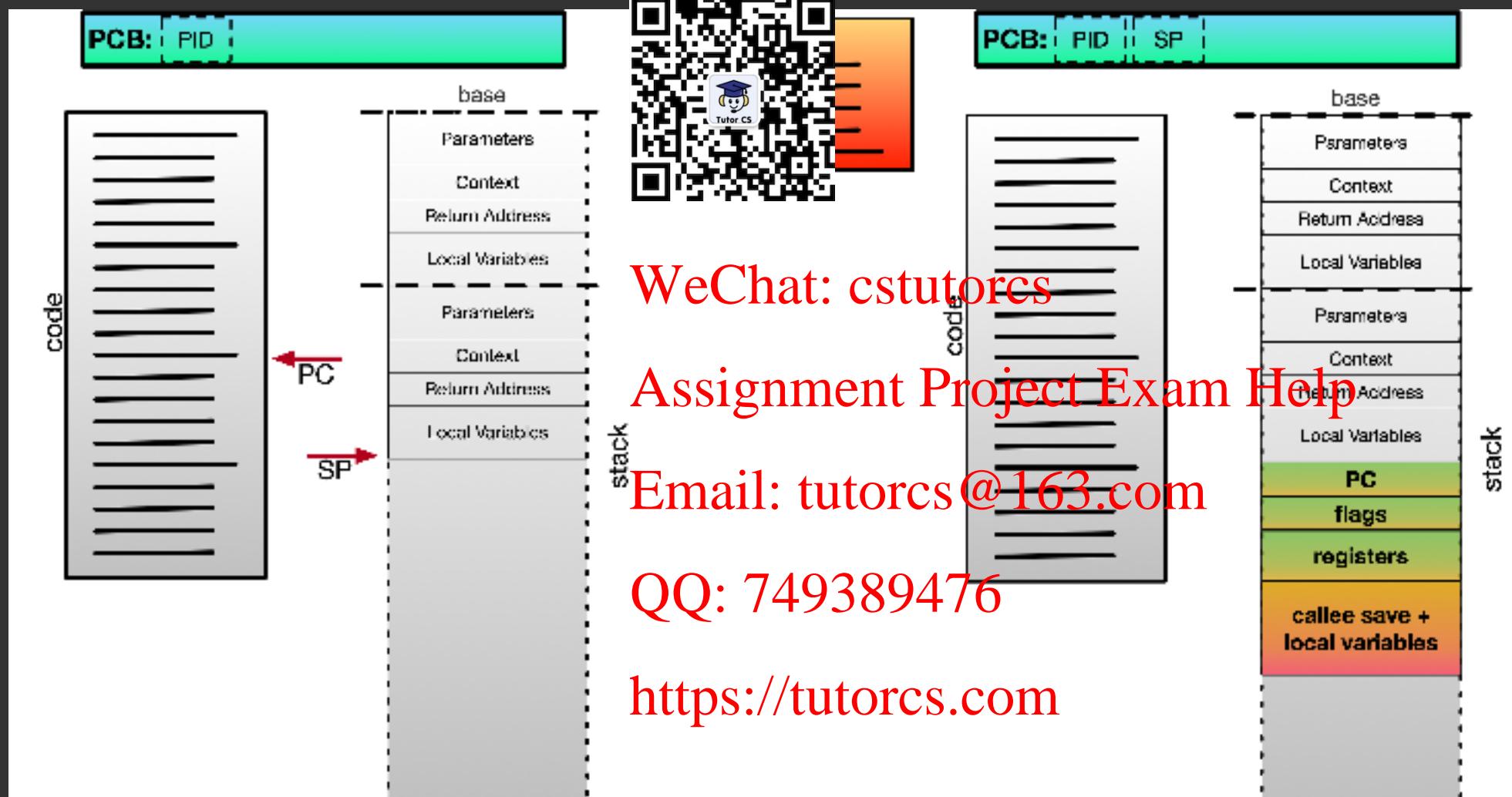
Email: tutorcs@163.com

QQ: 749389476

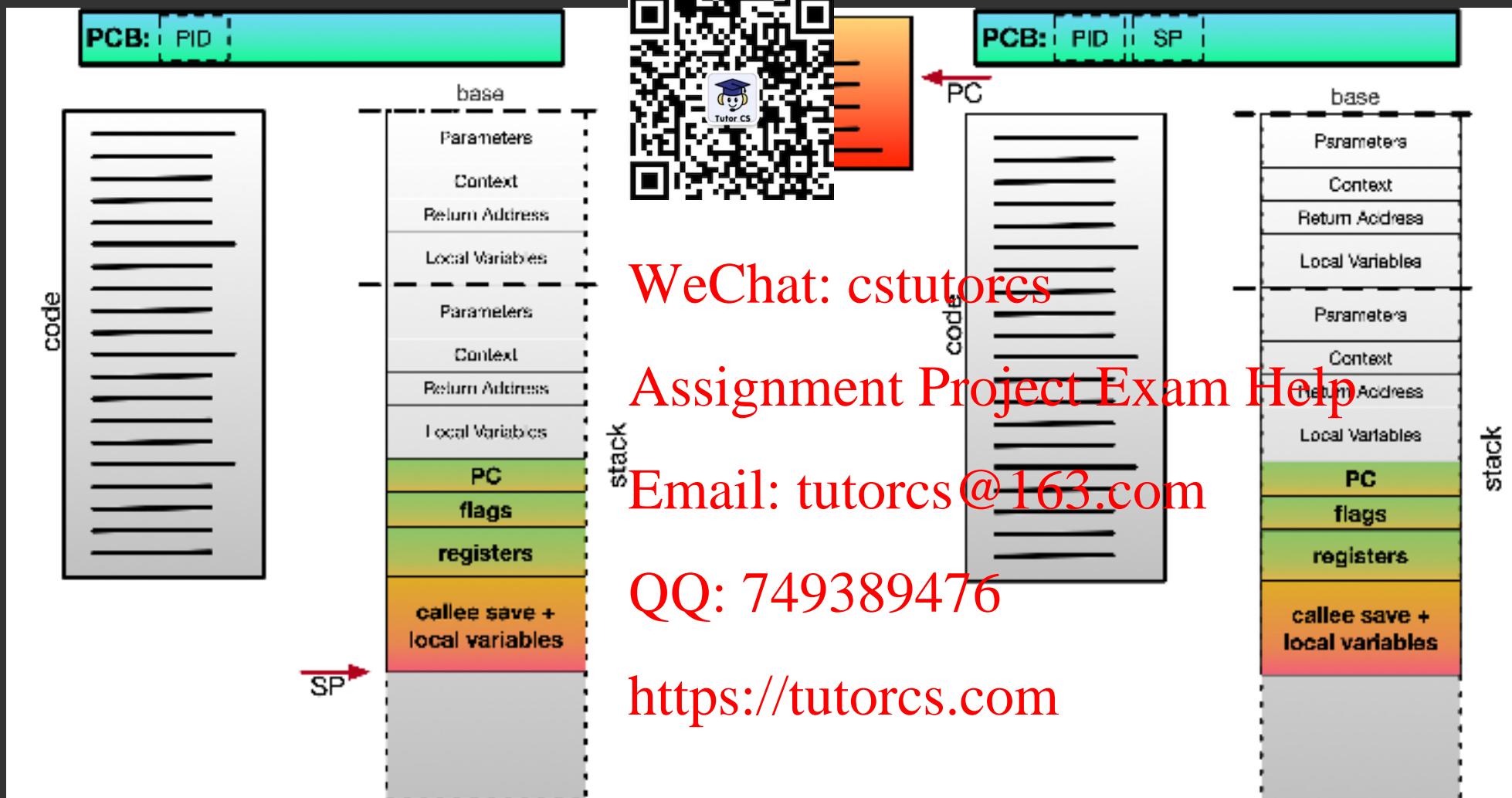
<https://tutorcs.com>



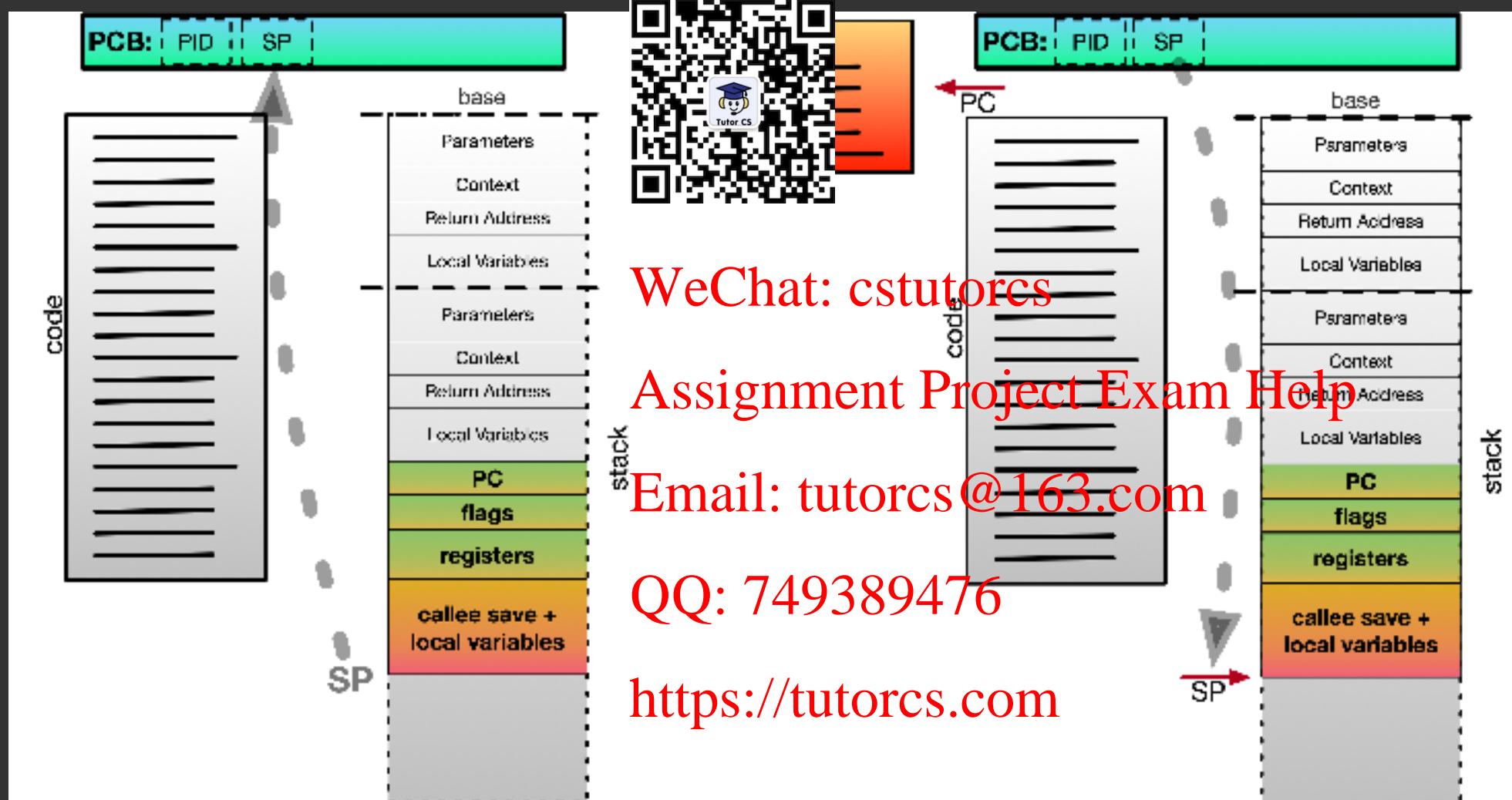
# Step 1: Before a context switch... 程序代写代做 CS编程辅导



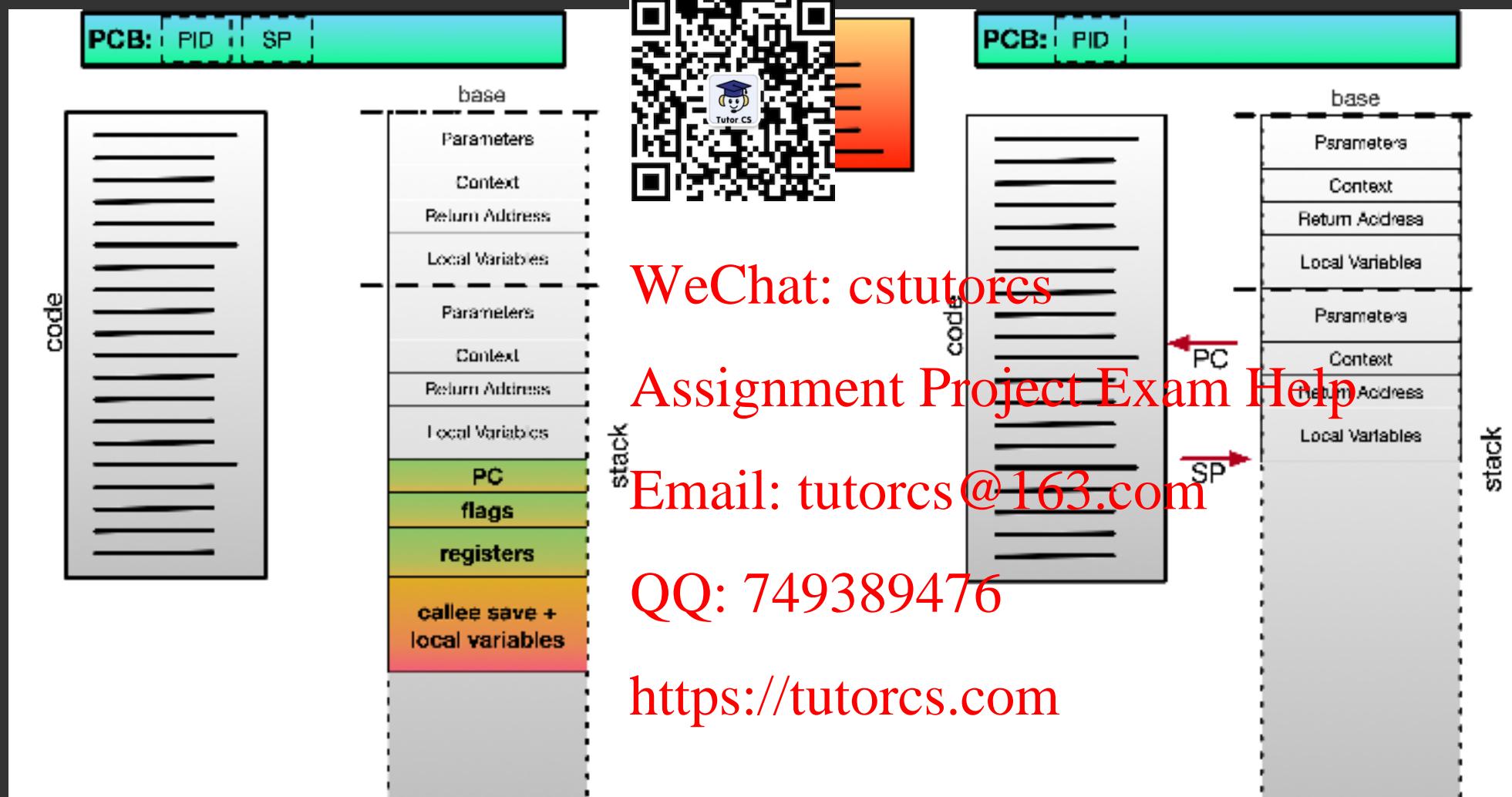
## Step 2: Interrupt! And switch to dispatcher... 程序代写代做 CS 编程辅导



# Step 3: Swapping to the new context... 程序代写代做 CS编程辅导



# Step 4: Returning from dispatcher... 程序代写代做 CS编程辅导



Will either program notice?

程序代写代做 CS编程辅导

Would a program notice if it's been read out and back?



Could anything go wrong here?

WeChat: cstutorcs

What if both programs *share data*?

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

## Concurrency and Synchronisation

---

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

What is the value of G?

```
int G = 0;
```

G = 1;

G = G + G;



G

G = G + G;

WeChat: cstutorcs

G = 3;

G = G + G;

What's the value of G at the end?

Assignment Project Exam Help

What's the *smallest* G that could be produced? What's the *largest*?

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## What is the value of G?

```
G: .word 0x00000000
```

```
ldr r4, =G  
mov r1, #1  
str r1, [r4]  
ldr r4, =G  
ldr r2, [r4]  
add r2, r2, r2  
str r2, [r4]
```

程序代写代做 CS编程辅导



```
ldr r4, =G  
mov r1, #2  
str r1, [r4]  
ldr r4, =G  
ldr r2, [r4]  
add r2, r2, r2  
str r2, [r4]
```

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476

What are the values of G that we can see now?

<https://tutorcs.com>

## The too-much-milk problem

程序代写代做 CS编程辅导

Fact 1: There are two roommates.



Fact 2: When a roommate gets home, they open the fridge. If there is no milk, they go out to get some.

WeChat: cstutorcs

Now what is going to happen? What is the solution?

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Solutions to the Too-Much-Milk Problem!

- a note? “I’ve gone to get milk!”
- a lock on the fridge door?
- two fridges? (ok now the analogy is going too far...)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Mutual Exclusion

程序代写代做 CS编程辅导

A “critical section” is a section where a program is accessing a shared resource.

We need a way to ensure that only one process can enter a critical section at a time.

This is called “mutual exclusion”.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# What happens without?

```
int count = 0;
```

```
for (i = 1; i <= 100; i++) {  
    count = count + 1;  
}
```

程序代写代做 CS编程辅导



```
for (i = 1; i <= 100; i++) {  
    count = count - 1;
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

What happens without?

程序代写代做 CS编程辅导



Count: .word 0x00000000

```
ldr r4, =Count  
mov r1, #1  
for_enter:  
    cmp r1, #100  
    bgt end_for_enter  
    ldr r2, [r4]  
    add r2, #1  
    str r2, [r4]  
    add r1, #1  
    b for_enter  
end_for_enter:
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
for_leave:  
    cmp r1, #100  
    bgt end_for_leave  
    ldr r2, [r4]  
    sub r2, #1  
    str r2, [r4]  
    add r1, #1  
    b for_leave  
end_for_leave:
```

# What does ARMv7-M give us? 程序代写代做 CS编程辅导

```
Count: .word 0x00000000
```

```
ldr r4, =Count  
mov r1, #1  
for_enter:  
    cmp r1, #100  
    bgt end_for_enter  
enter_strex_fail:  
    ldrex r2, [r4] ; tag [r4] as exclusive  
    add r2, #1  
    strex r2, [r4] ; only if untouched  
    cbnz r2, enter_strex_fail  
    add r1, #1  
    b for_enter  
end_for_enter:
```



```
ldr r4, =Count  
mov r1, #1
```

**for\_leave:**

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476  
<https://tutorcs.com>

```
ldrex r2, [r4] ; tag [r4] as exclusive  
sub r2, #1  
strex r2, [r4] ; only if untouched
```

cbnz r2, leave\_strex\_fail

add r1, #1

b for\_leave

**end\_for\_leave:**

Can we improve this?

程序代写代做 CS编程辅导

```
ldr r4, =Count  
mov r1, #1  
for_enter:  
    cmp r1, #100  
    bgt end_for_enter
```



```
ldr r4, =Count  
mov r1, #1  
for_enter:  
    cmp r1, #100  
    bgt end_for_leave
```

WeChat: cstutorcs

Negotiate who goes first

Assignment Project Exam Help

```
ldr r2, [r4]  
add r2, #1  
str r2, [r4]
```

Critical section

```
ldr r2, [r4]  
sub r2, #1  
str r2, [r4]
```

Critical section

QQ: 749389476

Indicate critical section completed

```
add r1, #1  
b for_enter  
end_for_enter:
```

```
add r1, #1  
b for_leave  
end_for_leave:
```

<https://tutorcs.com>

# Using a “lock” variable

程序代写代做 CS编程辅导

```
Count: .word 0x00000000
Lock: .word 0x00000000 ; #0 means unlocked

ldr r3, =Lock
ldr r4, =Count
mov r1, #1

for_enter:
    cmp r1, #100
    bgt end_for_enter

fail_enter:
    ldrex r0, [r3]
    cbnz r0, fail_enter ; if locked
    mov r0, #1           ; lock value
    strex r0, [r3]        ; try lock
    cbnz r0, fail_enter ; if touched
    dmb                 ; sync memory

    ldr r2, [r4]
    add r2, #1           Critical section
    str r2, [r4]

    dmb                 ; sync memory
    mov r0, #0           ; unlock value
    str r0, [r3]          ; unlock

    add r1, #1
    b for_enter

end_for_enter:
```

Any context switch  
needs to clear  
reservations

Critical section

for
 cmp r1, #100
 bgt end\_for\_leave

fail\_leave:
 ldrex r0, [r3]
 cbnz r0, fail\_leave ; if locked
 mov r0, #1 ; lock value
 strex r0, [r3] ; try lock
 cbnz r0, fail\_leave ; if touched
 dmb ; sync memory

 ldr r2, [r4]
 sub r2, #1 Critical section
 str r2, [r4]

 dmb ; sync memory
 mov r0, #0 ; unlock value
 str r0, [r3] ; unlock

 add r1, #1
 b for\_leave

end\_for\_leave:

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

end\_for\_leave:

page 312 of 481 (chapter 5, “synchronization” up to page 334)

程序代写代做 CS编程辅导



WeChat: cstutorcs

Alternatives to strex and ldrex  
Assignment Project Exam Help

---

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Atomic Operations

程序代写代做 CS编程辅导

An *atomic* operation is one that

happens completely, or not at all.

It's indivisible.

WeChat: cstutorcs

There's lots of operations we need to be atomic. (e.g., bank transfers)

Assignment Project Exam Help

(Photo by **Dan Meyers** on Unsplash)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Alternatives: Atomic Test-and-Set Operation

- a special instruction to *read a value* and *set as a new value* atomically.



```
type Flag is Natural range 0..1; C : Flag := 0;

task body Pi is
  L : Flag;
begin
  loop
    loop
      [L := C; C := 1];
      exit when L = 0;
      ----- change process
    end loop;
    ----- critical_section_i;
    C := 0;
  end loop;
end Pi;
```

WeChat: cstutorcs  
task body Pj is  
L : Flag;  
begin  
loop  
loop  
[L := C; C := 1];  
exit when L = 0;  
----- change process  
end loop;  
----- critical\_section\_j;  
C := 1;  
end loop;  
end Pj;

“keep reading the lock (and setting it to 1) until you find it open”

# Alternatives: Atomic Exchange Operation

- special instruction to atomically exchange values.
- similar idea...



```
type Flag is Natural range 0..1; C : Flag := 0;

task body Pi is
  L : Flag := 1;
begin
  loop
    loop
      [Temp := L; L := C; C := Temp];
      exit when L = 0;
      ----- change process
    end loop;
    ----- critical_section_i;
    L := 1; C := 0;
  end loop;
end Pi;
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

[Temp := L; L := C; C := Temp];

QQ: 749389476

<https://tutorcs.com>

```
L := 1; C := 0;
end loop;
end Pj;
```

# ARMv7-M Memory Cell Reservation 程序代写代做 CS编程辅导

This is what we have! Also called “load/store-conditional”.



```
type Flag is Natural range 0..1; C : Flag := 0;  
  
task body Pi is  
  L : Flag;  
begin  
  loop  
    loop  
      L := C; C := 1;  
      exit when Untouched and L = 0;  
      ----- change process  
    end loop;  
    ----- critical_section_i;  
    C := 0;  
  end loop;  
end Pi;
```

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476  
<https://tutorcs.com>

```
task body Pj is  
begin  
  loop  
    loop  
      L := C; C := 1;  
      exit when Untouched and L = 0;  
      ----- change process  
    end loop;  
    ----- critical_section_j;  
    C := 0;  
  end loop;  
end Pj;
```

# Semaphores

程序代写代做 CS编程辅导

A **semaphore** (Dijkstra, 1968) is  
more generally.



extension of our *lock* to handle common resources

We define a *semaphore variable* ( $S$ ), and two operations:

WeChat: cstutorcs

**Wait( $S$ )**: if  $S > 0$ , then  $S := S - 1$  and continue, else wait

Assignment Project Exam Help

**Signal( $S$ )**:  $S := S + 1$ , tell someone waiting to try again

Email: tutorcs@163.com

So  $S$  could start at 1 (binary) or a higher number.

QQ: 749389476

<https://tutorcs.com>

```
Count: .word 0x00000000  
Sema: .word 0x00000001
```

```
ldr r3, =Sema  
ldr r4, =Count  
mov r1, #1  
for_enter:  
    cmp r1, #100  
    bgt end_for_enter
```

Any context switch  
needs to clear  
resources

**wait\_1:**

```
ldrex r0, [r3]  
cbz r0, wait_1 ; if Semaphore = 0  
sub r0, #1 ; dec Semaphore  
strex r0, [r3] ; try update  
cbnz r0, wait_1 ; if touched  
dmb ; sync memory
```

...

**Critical section****signal\_1:**

```
ldrex r0, [r3]  
add r0, #1 ; inc Semaphore  
strex r0, [r3] ; try update  
cbnz r0, signal_1 ; if touched  
dmb ; sync memory
```

```
add r1, #1  
b for_enter
```

**end\_for\_enter:**

```
ldr r3, =Sema  
ldr r4, =Count  
mov r1, #1  
for_leave:  
    cmp r1, #100  
    bgt end_for_leave
```

**wait\_2:**

```
ldrex r0, [r3]  
cbz r0, wait_2 ; if Semaphore = 0  
sub r0, #1 ; dec Semaphore  
strex r0, [r3] ; try update  
cbnz r0, wait_2 ; if touched  
dmb ; sync memory
```

...

**Critical section****signal\_2:**

```
ldrex r0, [r3]  
add r0, #1 ; inc Semaphore  
strex r0, [r3] ; try update  
cbnz r0, signal_2 ; if touched  
dmb ; sync memory
```

```
add r1, #1  
b for_leave
```

**end\_for\_leave:**

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

This happens all the time!

程序代写代做 CS编程辅导

Concurrency is not an edge case.



ed all the time.

Multi-task, and multi-CPU systems require hardware support for concurrency.

WeChat: cstutorcs

Now you know the *hardware operations* that can support synchronisation implementations in your code.

Assignment Project Exam Help

Email: tutorcs@163.com

In general, **higher abstractions** and **safer** solutions are provided; learn more in **COMP2310!**

QQ: 749389476

<https://tutorcs.com>

Let's try it on a microbit: LDREX

程序代写代做 CS编程辅导

```
ldr r0, =label_in_data  
ldrex r1, [r0]
```



ion

ldrex loads r1 with the memory that r0 is pointing to, and sets the “local exclusive monitor” to exclusive.

WeChat: cstutorcs

Assignment Project Exam Help

It doesn't do any checking of the exclusive monitor before doing so!

Email: tutorcs@163.com

The local exclusive monitor is just 1-bit! Either “open” or “exclusive”.

QQ: 749389476

(Multi-processor systems also have a “global exclusive monitor”... not covered here!)

<https://tutorcs.com>

```
ldr r0, =label_in_da  
mov r1, 5  
strex r2, r1, [r0]  
cmp r2, 0  
bne do_something_to_recover
```



WeChat: cstutorcs

strex tries to store r1 in the memory that r0 is pointing to, but checks the exclusive monitor first.

Assignment Project Exam Help

Email: tutorcs@163.com

If the store is allowed, r2 is set to 0; if it fails, then r2 is set to 1.

QQ: 749389476

Then what should we do?

<https://tutorcs.com>

# When can a `strex` actually fail? 程序代写代做 CS 编程辅导

Need to look in the ARMv7-M reference manual (Section A3.4 “Synchronisation and Semaphores”)



1. If address of `strex` is tagged exclusive in the local monitor; then store takes place (woo hoo!)

WeChat: cstutorcs

2. If address of `strex` is NOT tagged exclusive; then it is “implementation defined” whether the store takes place (????).

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Local Exclusive Monitor

程序代写代做 CS编程辅导

“Any `ldrex` operation updates address to the most significant bits of the address... used for the operation” (ARM reference manual)



Note that `clrex` always clears the monitor, and that interrupt handlers run `clrex`!

## Further Reading

程序代写代做 CS 编程辅导

G. Taubenfeld. Concurrent programs for mutual exclusion (1965; Dijkstra), Encyclopedia of Algorithms (2008) 188–191 <https://faculty.idc.ac.il/gadi/MyPapers/2008T-mutex.pdf>



Wikipedia: Load-link and store conditional

WeChat: cstutorcs

Wikipedia: Semaphores

Assignment Project Exam Help

ARM Synchronization Primitives

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

Mutual Exclusion Non-Solutions (Youtube)

QQ: 749389476

<https://tutorcs.com>

## Making a `millis` function

程序代写代做 CS编程辅导

Is there a smarter way to measure



the microbit?

How about a function that tells you how much time has passed?

WeChat: cstutorcs

On an Arduino or in `p5` or Processing you can call `millis()` which returns how long your program has been running in milliseconds. How does that work?

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Millis Function

程序代写代做 CS编程辅导

```
@ returns: r0, number of milliseconds
.type millis, %function
millis:
    ldr r0, =milliseconds_count
    ldr r0, [r0]

    bx lr
.size millis, . - millis

.global SysTick_Handler
.type SysTick_Handler, %function
SysTick_Handler:
    @ update the milliseconds count.
    ldr r0, =milliseconds_count
    ldr r1, [r0]
    add r1, r1, #1
    str r1, [r0]
    bx lr

.data
milliseconds_count:
.word 0x0
```



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Setup

## 程序代写代做 CS编程辅导

```
@ enable SysTick - lower three bits  
ldr r0, =ADR_SYST_CSR  
ldr r1, =0b111 @ (enable systick,  
str r1, [r0]
```



SR (SysTick Control and Status Register)

interrupt, set CPU as clock source)

```
@ Store a reload value in SYST_RVR (Reload Value Register)  
ldr r0, =ADR_SYST_RVR  
ldr r1, =64000 @ 1ms @ 64MHz  
str r1, [r0]
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Questions

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>