

ระบบการเตือนข้อความการนัดหมาย

จัดทำโดย

5570984921 น.ส. นิธิกานต์ คำจอร์

เสนอ

ศาสตราจารย์ ดร. ประภาศ จงสฤษดิ์วัฒนา

รายวิชานี้เป็นส่วนหนึ่งของวิชา Embedded and Real-Time system

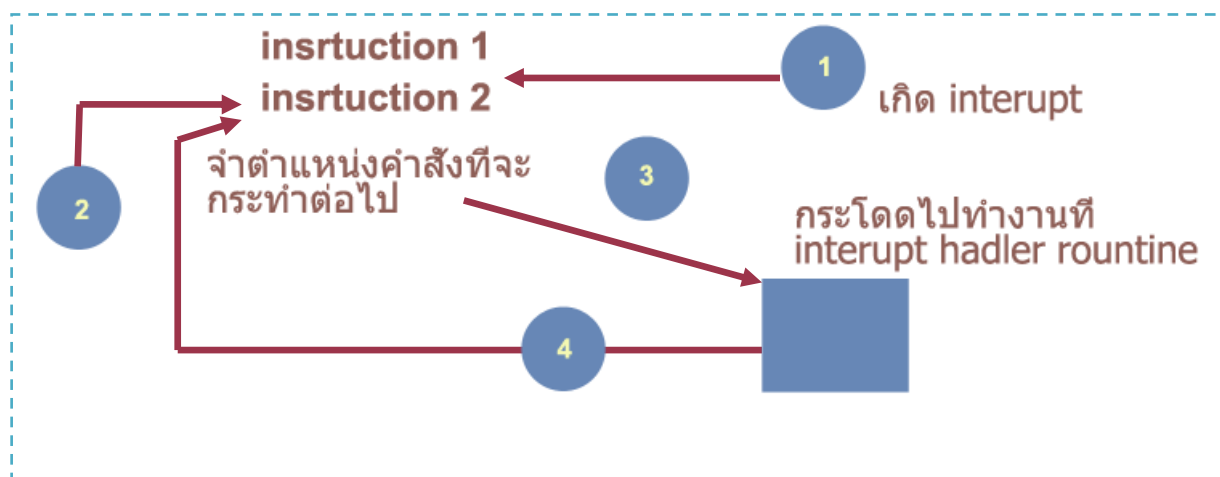


Suppose

1. เพื่อให้มันเกิดทักษะในการเขียนโปรแกรมภาษา Assembly มากขึ้น
2. เพื่อให้มันใส่ใจหลักการของ Timer Interrupt

Design

Interrupt มีสองประเภท คือ software interrupt (หรือ trap) และ hardware interrupt ไม่ว่าจะเป็น interrupt แบบใด จะมีขั้นตอนการจัดการกับ interrupt เหล่านั้นด้วยการจดจำตำแหน่งของคำสั่งที่จะ execute ถัดไป ไว้บนกองซ้อน (push CS, push IP) แล้วกระโดดไปยัง interrupt handler routine (เปลี่ยนค่าใน CS, IP) เพื่อบริการ interrupt ดังกล่าว เมื่อบริการ interrupt handler routine เสร็จ ก็จะกลับมาทำงานที่คำสั่งที่ตำแหน่งซึ่งได้เคยจดจำไว้ (pop IP, pop CS) ดังรูปภาพด้านล่าง



Motivation

แนวความคิดในการเขียนโปรแกรมนี้นั้นขึ้นมาจากว่า ณ ปัจจุบันนี้การทำงานต้องแข่งขันกับเวลา ในวันหนึ่งๆ ผู้คนอาจต้องมีการนัดหมาย หรือมีกิจกรรมต่างๆ มากมาย และวิธีการหนึ่งที่จะสามารถช่วยเตือนความจำได้ก็คือ การตั้งเวลาเตือนความจำพร้อมข้อความเล็กน้อยเพื่อบอกให้เราได้รู้ว่า เรากำลังมีนัดอะไรอยู่





Code of the demo system

หลักการ

1. เริ่มตั้งค่า วินาทีเป็น 0 จากนั้น ในการ interrupt แต่ละครั้งจะทำการเพิ่มค่า วินาทีทีละหนึ่งไปเรื่อยๆ
2. ทำการสร้างเงื่อนไขว่า ถ้า "วินาที mod ด้วย 60 ลงตัว" แสดงว่าวินาทีที่วิ่งครบรอบ 1 นาทีแล้ว
3. เพิ่มค่านาทีทีละหนึ่ง ทุกครั้งหลังการ mod ของวินาทีด้วย 60 ลงตัว
4. ในเงื่อนไขของวินาที ถ้าวินาทีได้ถูกเพิ่มถึงค่า 60 นาที นั้นแสดงว่า คำนั้นครบ 1 ชั่วโมงแล้ว
5. เพิ่มค่าชั่วโมงทีละหนึ่ง ทุกครั้งหลังการ mod ของนาที่ด้วย 60 ลงตัว
6. และเมื่อชั่วโมงบวกกันจนครบ 24 ชั่วโมงถือว่าจบการทำงาน



ตัวอย่างโค้ด

```
.symbol
x 1
y 2
z 3

.code 0
; setup interrupt
mov r1 #int_clock
st r1 1000 ; int vector
mov r1 #1
trap r1 #12 ; enable int
mov r1 #1000
trap r1 #10 ; set timer1 =
1000
mov r1 #100
trap r1 #11 ; set div = 100

; main program
mov x #0
mov y #0
mov z #1

:while
mod r4 z #60
eq r5 r4 #0
jmp xxxx
jt r5 while2
jmp while

:while2
add y y #1
mov r7 #y
mod r8 r7 #60
eq r9 r8 #0
jt r9 while3
jmp while

:while3
add x x #1
mov r10 #x
eq r11 r10 #24
jt r11 exit3
jmp while

:xxxx
mov r5 #z
mov r6 r4

:exit3
trap r0 #0

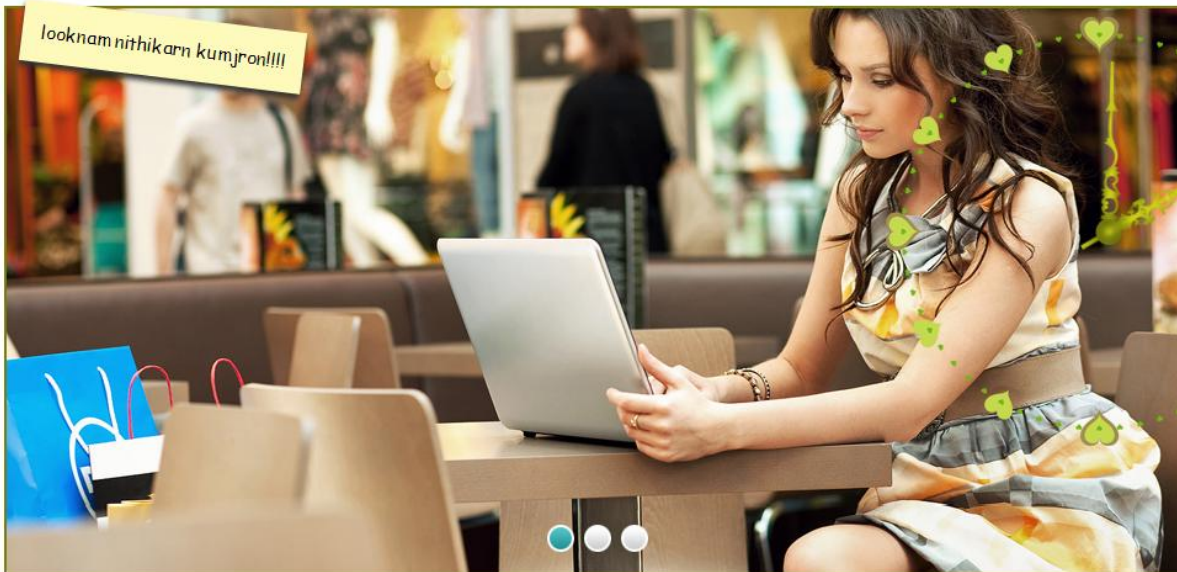
:int_clock
add z z #1

.data 100
0
.end
```



Result

ผลลัพธ์ที่ได้ ก็จะเป็นข้อความที่เราได้ set ไว้ ตามเวลาที่กำหนดก็จะแสดงผลออกมา



Code:

```
<script >
var wait_time=8;
window.onload=function(){
  var vela=setInterval("decrease_num()",1000);
  setTimeout("clearInterval("+vela+")",(wait_time+1)*1000);
}
function decrease_num(){
  wait_time--;
  if(wait_time==0){
    document.getElementById('show_area').style.display = "block";
    var show_place=document.getElementById('show_text');
    show_place.innerHTML= "looknam nithikarn kumjron!!!!";
  }
}
</script>
```

Tools

1. Assembler package new version as23-1.zip
2. Embed System simulator written in Javascript ems.js

