

PRÁCTICA 1 DE ARQUITECTURA DE COMPUTADORES:

Diseño de un BenchMark sintético en Ensamblador y Cálculo de MIPS.

2º Ingeniería Informática - UHU Arquitectura de Computadores Profesor: Juan Manuel Ponce Real Curso: 2020/2021

Alumnos Alba Márquez Rodríguez Alberto Fernández Merchán

Programa Pinta_Pixel

```
.MODEL small
.DATA
    color db 01h
    contador dw 00h
   fin_linea dw 00h
                                    \rightarrow 140h x 2 grafico = 200 filas x 320 colum
   npant dw 00h
   aux dw 00h
    . CODE
        MOV AX,seg color
MOV DS,AX
MOV fin_linea, 00h
                                            establecer el modo de pantalla en modo grafco
                            ; int
                                      9 instrucciones
                                      fuera del bucle
                                                                memori 11 instrucciones fuera
        MOV AX, ØAØØØh
MOV ES, AX
                                             la direccian de
                                                                               del bucle
         MOV SI. 280h
                           ;no se pita en las dos primeras
                AH,
                     4ch
21h
          MOV
                                     2 instrucciones fuera del bucle
                INT
    END
```

Programa Pinta_Pixel

npant

Contador que muestra en las primeras líneas un píxel por cada pantalla empezada

```
color, 01h
pinta_pixel
          00h
reset_color
```

Porque la

Reinicio

pantalla se ha llenado

Contador fin de línea

Permanece en el bucle mientras no se llegue al fin de linea

pinta_pixel: MOV AL, MOV ES:[ES [SI] AL fin_linea,02h fin_linea, 13F JBE pinta pixel

;hacemos salto de linea: ADD SI,280h ;500h MOV fin_Linea, 00h

Saltar una línea = ADD SI, 140h Para que se distinga mejor

saltamos dos líneas

Salto de línea

Parte del código que se ejecuta cuando se salta una línea

Cuando se llena una pantalla

incremento contador para saber cuando termina contador, Pin CMP contador, 13Fh JAE cambia_color

JMP pinta_pixel

Contador

Controla que se llene la pantalla entera, si se llena, cambia el color v se reinicia todo

Programa Pinta_Pixel

```
MOV color, 01h
                              pinta_pixel
                             color:
                             contador,
vux, SI
                              color, 01h
                                                           Si fin de pantalla:
                                                            + 9 instrucciones
                              SI, npant
ES:[SI], AL
                                                            Si reinicia color:
                                                           +9+3 instrucciones
                              reset_color
    pinta_pixel:
MOV AL,
MOV ES:[
3901234567890123456
                ES:[SI]
                                                      Bucle principal
                     fin_linea,02h
fin_linea, 13Fh
                                                      6 instrucciones
          JBE pinta_pixel
          ;hacemos salto de
ADD <mark>SI</mark>,280h ;500h
MOV fin_linea, 00h
                                     lineat
                                                Si fin de línea:
                                               +5 instrucciones
            incremento contador
                                                  saber cuando termina
                contador, 01h
                contador, 13Fh
                                                  Si fin de pantalla:
                cambia_color
                                                 +5+1 instrucciones
          JMP pinta_pixel
```

Programa Residente: R.S.I.

```
.MODEL SMALL
. CODE
                                                             Contador de
     org 100h
Programa_Int:
JMP Reside
                                                             Interrupciones.
                                                             Sale de la RSI cuando
     contador_int db 0
                                                             llega a <u>91</u> interrupciones.
Rutina_Servicio PROC
     INC contador_int
     CMP contador_int,
                                181.2 interrupciones / 1 segundo --> 91 interrupciones / 5 segundos
     JNE fin
              4ch
                                      6 instrucciones \cdot 90 veces + 6 instrucciones \cdot 1 vez = 546
fin:
                                      instrucciones en 5 segundos
ENDP
Reside:
    MOV DX, OFFSET Rutina_S

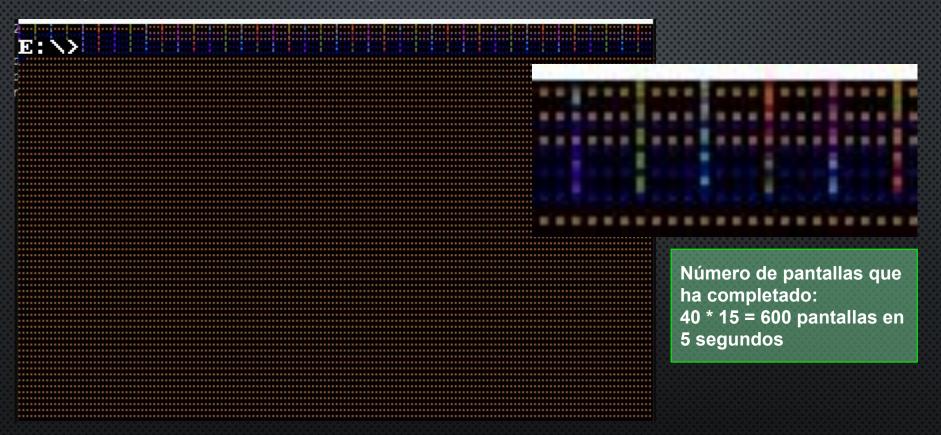
MOV AX, Ø

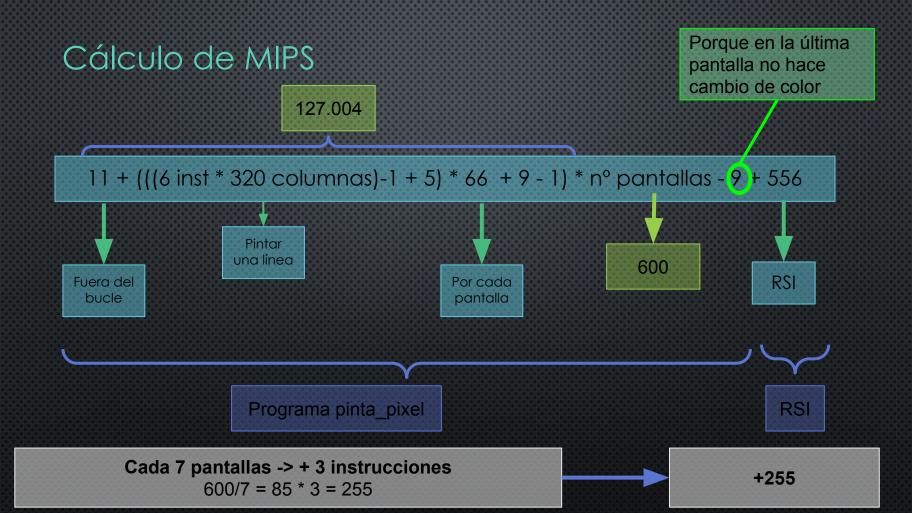
MOV ES, AX; ES vale Ø.

MOV SI, 1Ch*4

CLI ; BIEST
              OFFSET Rutina_Servicio
                                                                                            instrucciones
     STI
                      residente desde la etiqueta hacia atras (rutina)
END Programa_Int
```

Ejecución del Código





Cálculo de MIPS

