

# Computación



GNU GRUB 1.99-21

Memory Test (memtest86)

M.1|\|t\$ 10 128 bit [dev\sda1]

P2: HL-dt-st-486

NVME. m.2 Debian GNU/Linux 3.2.0

UEFI: ADATA UFD

Boot selecte

Aptec RAID BIOS v7.4

CPU : AMD QX3700+ @7.99hz

128 bit mode enabled

SMC :: smc ReadyKeyfiction: 0x0000\*\*\*\*\*

Press DEL to run Setup

3584825480 GB OK

C:\>

Disk /dev/sda 393912.1 GB

255 heads, 63 sectors/track

Units = sectors of 1

I/O size

Device	Boot	Start	System
/dev/sda1		2048	HPFS/NTFS
/dev/sda2		408674	exFAT
/dev/sda3		522512	Extended

[ok] Waiting for /dev to be populated  
192.168.1.1 Debian base

[ok] Activating swap...done

[..] root file system

/dev/mmcblsals: clean, 8483248 files

[!t] Cleaning up temp files

[ok] mounting local filesystem

[\*\*] configuring network interface

[ok] 855it(0) up ALSA..done

[ok] Mount disk 1

[..] NTFS kernel

/dev/mmcblsals: clean, 8483248 files

# Estructuras de control

## Sentencia IF() Bloque

La forma IF() sentencia que vimos en la clase anterior sólo permite que se use una sola orden y no más

IF(algo) sentencia (gato número muy mala idea)

La sentencia IF bloque permite desarrollar la orden IF() usando muchas sentencias, todas las que sean necesarias.

Esto tiene una ventaja apreciable

# Forma de construirse

```
IF (algo) then
```

```
...
```

```
...
```

```
...
```

```
ELSEIF(algo2) then
```

```
...
```

```
...
```

```
...
```

```
ELSEIF(algo3) then
```

```
...
```

```
...
```

```
...
```

```
ELSE
```

```
...
```

```
..
```

```
...
```

```
ENDIF
```

# Ejemplo de como funciona

- Función a trazos

Supongamos que en alguna parte de un programa, deberíamos calcular la siguiente función a trazos:

$$f(x) = \begin{cases} 0 & \text{si } x \leq 30 \\ x - 30 & \text{si } 30 < x < 60 \\ 30 & \text{si } 60 \leq x \end{cases} \quad (7.1)$$

La parte del programa que hace este trabajo tendría la siguiente forma:

⋮

**IF (X.LE.30) THEN**

**F = 0**

**ELSEIF(X.GT.30.AND.X.LT.60) THEN**

**F = X - 30**

**ELSE**

**F = 30**

**ENDIF**

⋮

# Raíces de un polinomio de 2do grado

En una ecuación de segundo grado  $Ax^2 + Bx + C = 0$ , tengo que calcular el discriminante  $B^2 - 4AC$  y según si es positivo, negativo o igual cero tendré soluciones muy diferentes (2 soluciones reales, 2 complejas o una sola, en cada caso)

Solución: 
$$X_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

Recordar que si quiero convertir un número a su forma de variable compleja tengo que hacer:

Número complejo=COMPLEX(parte real (real), parte imaginaria (real))

Logical Z3

$a=3.5$  se puede pensar que  $a=3.5+ 0 i$

$Z3=\text{complex}(3.5,0)$

Ahora Z3 es un número complejo para la computadora

```
program cuad
complex z1,z2

read(*,*) a,b,c

disc=b*b-4*a*c

if(disc.lt.0) then
    z1=(-b+sqrt(complex(disc,0)))/(2*a)
    z2=(-b-sqrt(complex(disc,0)))/(2*a)
    write(*,*)'raices complejas:',z1,z2

elseif(disc.eq.0) then
    x=-b/(2*a)
    write(*,*)' el disc. es cero, x=',x

else
    x1=(-b+sqrt(disc))/(2*a)}
    x2=(-b-sqrt(disc))/(2*a)}
    write(*,*) x1,x2}

endif
end
```



Vemos el programa corriendo en una terminal

Busquemos las raíces de:

$$X^2 - X - 6 = 0$$

$$X^2 + 2X + 1 = 0$$

$$X^2 + 4X + 7 = 0$$

$$X^2 + X + 1 = 0$$