

---

# ***Programación, compilación e depuración***

***GPUL***

Xabier Rodríguez Calvar



- Programación
- Compilación (Make)
- Depuración (GDB)
- Utilización de Xemacs como *IDE*

- `program.pas:`  
`program a_program;`  
  
`begin`  
    `writeln ( 'Hello World!' );`  
`end.`
- **Compilación:**
  - `$ fpc program.pas`
  - `$ gpc -o program program.pas`

- hello.pas:  
unit hello;  
  
interface  
procedure print\_hello;  
  
implementation  
procedure print\_hello;  
begin  
    writeln('Hello World!');  
end; { print\_hello }  
end.

- `program.pas:`  
`program a_program;`  
  
`uses hello;`  
  
`begin`  
    `print_hello;`  
`end.`
- **Compilación:**
  - `$ fpc program.pas`
  - `$ gpc --automake -o program program.pas`

- Xeito moi sinxelo. Problemas:
  - Outro compilador
  - Proxectos máis complexos.
- Dúas etapas na compilación:
  - Compilación (`program.o hello.o`)
  - Enlazado (`program`).

- Ferramenta `Make`: encárgase de controlar que só se compilen os últimos ficheiros modificados.
- Ficheiro `Makefile`:
  - *Targets*
  - Prerrequisitos
  - Instruccions
- Compilación:  
\$ `make`

- Ficheiro Makefile:

```
program: program.o hello.o
        gpc -g -o program program.o hello.o
```

```
hello.gpi: hello.o
```

```
hello.o: hello.pas
        gpc -g -c -o hello.o hello.pas
```

```
program.o: program.pas hello.gpi
        gpc -g -c -o program.o program.pas
```

```
clean:
```

```
rm -f hello.ppu program program.o hello.o \
*~ hello.gpi
```



- Engadimos función a hello.pas:

```
procedure print_hello_n_times(n : integer);  
var  
    i : integer;  
    hello_strings : Array[1..5] of String;  
begin  
    for i:=1 to n do begin  
        hello_strings[i] := 'Hello World!';  
    end;  
    for i:=1 to n do begin  
        writeln(hello_strings[i]);  
    end;  
end; { print_hello_n_times }
```

- Engadimos función a `program.pas`:

```
program a_program;
```

```
uses hello;
```

```
begin
```

```
    print_hello;
```

```
    writeln ( '-----' );
```

```
    print_hello_n_times(7);
```

```
end.
```

- Isto dará lugar a un erro que trataremos de localizar co GDB.
- Comandos útiles:
  - `set args`: establece os parámetros do programa
  - `run`: executa o programa
  - `break`: coloca un *breakpoint*
  - `delete`: quita un *breakpoint*
  - `cont`: fai que un programa continue
  - `next`: executa a seguinte instrucción no código
  - `step`: executa a seguinte instrucción máquina

- Comandos útiles:
  - `set`: troca valores de variábeis
  - `print`: escribe puntualmente o valor dunha variábel
  - `display`: vai escribindo o valor das variábeis según se executa o programa
  - `quit`: sae do programa (*Control + D*)

- Edición.
- Compilación (compile).
- Depuración (M-x gdb).

# ¿Preguntas?

A aprender **GNU/Linux!**