
Editores de texto e outras ferramentas de programación

Xornadas de Introducción a GNU/Linux (GPUL)

Xabier Rodríguez Calvar



- Editores de texto
 - Nano (Pico)
 - FTE
 - Vi(m)
 - GNU/Emacs
- Compilación (Make)
- Depuración (GDB)
- Utilización de GNU/Emacs como *IDE*

- Serven para trocar o contido dun ficheiro
- Moi importante en tarefas de programación
 - Os ficheiros de código fonte son texto
- Visores: permiten ver o contido dun ficheiro, pero non editalo:
 - most
 - less
 - more
- Outro editores:
 - *Hexadecimais*

- Editor básico
 - Edición
 - *Cut & paste*
 - Buscar e reemplazar
 - *Syntax highlighting* e indentación automática?
 - Xeito escuro
- Contrapartida Pico non libre
 - Máis básico aínda

- Editor máis avanzado
 - Edición
 - *Cut & paste*
 - Buscar e reemplazar
 - Gráfico e en terminal
 - *Syntax highlighting* e indentación automática
 - *Buffers*
- Problema coas cuotas!

- Editor básico
 - Edición
 - *Cut & paste*
 - Buscar e reemplazar
- En todos os sistemas
- Modo *edición e comando*
- Contrapartida Vim
 - Moito máis completo

- O editor
 - Todo
- Contrapartida XEmacs
- Aprender a tocar o piano

- Manexo de ficheiros e buffers
 - Buscas e substitucións
 - Comentarios automáticos
 - Utilización de comandos
 - Macros
-
- Ollo coa tecla META

- `program.pas:`
`program a_program;`

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

- program.c:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```

- Compilación:

- \$ gcc -c program.c

- Dúas etapas na compilación:
 - Compilación → program.o
 - `$ gcc -c program.c`
 - Ligado → program
 - `$ gcc -o program program.o`

- hello.c:

```
#include <stdio.h>
```

```
void hello()  
{  
    printf("Hello World!");  
}
```

- hello.h:

```
void hello(void);
```

- **Compilación:**

- `$ gcc -c hello.c → hello.o`

- `program.c`:

```
#include "hello.h"
```

```
int main()  
{  
    hello();  
    return 0;  
}
```

- **Compilación:**

- `$ gcc -c program.c → program.o`

- **Ligado:**

- `$ gcc -o program program.o hello.o`

- Xeito moi sinxelo → Problemas:
 - Outro compilador
 - Proxectos máis complexos
 - Automake e Autoconf
- Dúas etapas na compilación:
 - Compilación (`program.o hello.o`)
 - Ligado (`program`)

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

- Fichero Makefile:

```
program: program.o hello.o
```

```
    gcc -o program hello.o program.o
```

```
hello.o: hello.c hello.h
```

```
    gcc -Wall -c hello.c
```

```
program.o: program.c hello.h
```

```
    gcc -Wall -c program.c
```

```
clean:
```

```
    rm -f *~ hello.o program.o program
```


- `program.pas`:
`program a_program;`

`uses hello;`

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

- Ficheiro Makefile:

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

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

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

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

```
clean:
```

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

Consellos de alumno vello

- Limpar antes de levar
 - `$ make clean`
- Comprimir
 - `$ (g)tar cvfz codigo.tgz codigo`
- Descomprimir
 - `$ rm -r codigo`
 - `$ (g)tar xvfz codigo.tgz → codigo`

Consellos de alumno vello (II)

- Fin de liña
 - `$ fromdos codigo.c`
- Indentación automática
 - `$ indent codigo.c`
 - GNU/Emacs → `indent-region`

- Engadimos función a `program.c`:

```
void hello_n(int n)
{
    int i;
    for (i=0; i < n; i++)
        {
            hello();
        }
}
```

- Para seguir a ejecución dun programa usamos o GDB
 - Compilación con `-g`
 - `$ gcc -g -c hello.c`
- 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 continúe

- Comandos útiles:
 - `next`: executa a seguinte instrucción no código
 - `step`: executa a seguinte instrucción máquina
 - `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 (`M-x compile`)
- Depuración (`M-x gdb`)

- Control de versiones
 - CVS
 - Subversion
 - ...

Más documentación

- GNU/Emacs
 - `www.gnu.org/software/emacs`
- Make
 - `www.gnu.org/software/make`
- GDB
 - `www.gnu.org/software/gdb`
- Google :-)

- Referencias rápidas de `www.refcards.com`
 - GNU/Emacs
 - GDB
- A miña configuración de GNU/Emacs
 - `emacs_conf.tgz` → `.emacs`
 - Ollo, oculto
- Código de exemplo
- Código da presentación

Preguntas?