

# Clase Auxiliar II

Prof: L. Mateu

Aux: M. Leyton

9 de Agosto del 2004

## 1. Uso de Makefiles

Un Makefile es un archivo que permite ejecutar alguna o mas acciones siempre y cuando se cumplan las condiciones lógicas definidas para esa acción.

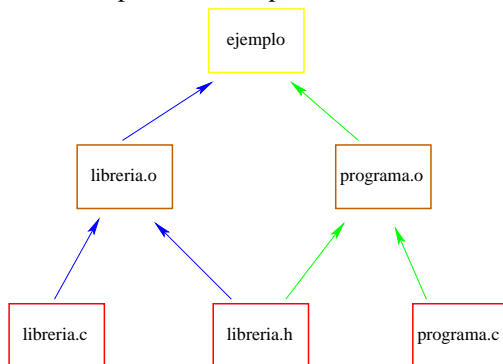
En general los Makefiles se utilizan para compilar archivos, como veremos a continuación, pero su uso no está limitado a esta funcionalidad.

Un archivo Makefile se compone principalmente de reglas. Cada regla tiene ninguna o mas condiciones (dependencias), y cero o mas instrucciones. Para poder ejecutar las instrucciones de una regla, primero deben satisfacerse las condiciones de la misma.

Veamos, por ejemplo el siguiente caso. Tenemos un `programa.c` que utiliza una `libreria.c` mediante el archivo `libreria.h`. La forma de compilar esto seria:

```
gcc -c libreria.c
gcc -c programa.c
gcc libreria.o programa.o -o ejemplo
```

Gráficamente, lo que estamos haciendo es crear un árbol donde los nodos padre dependen de la existencia de los hijos antes de poder ser compilados.



Las dependencias anteriores, junto con las acciones de compilar, pueden ser reguladas mediante un Makefile como el que se muestra a continuación.

```
default: ejemplo
ejemplo: programa.o libreria.o
    gcc programa.o libreria.o -o ejemplo
libreria.o: libreria.c libreria.h
    gcc -c libreria.c
programa.o: programa.c libreria.h
    gcc -c programa.c
clean:
    rm -fr *.o
cleanall: clean
    rm -fr ejemplo
```

Cada línea del archivo que contiene una palabra, seguida de un `:` es considerada una nueva regla. Sus condiciones (dependencias) son las palabras que se encuentran después de los dos puntos. Estas palabras pueden representar otras reglas del mismo Makefile o archivos que deben existir en la misma carpeta. Las líneas siguientes contendrán las instrucciones que deben ser ejecutadas al satisfacer esa regla.

Aquí la regla defecto se satisface si y sólo si se cumple alguna de las siguientes condiciones:

- Existe alguna regla llamada `ejemplo`, y esta se satisface recursivamente.
- Existe un archivo llamado `ejemplo` y su fecha de modificación es más reciente que todas sus dependencias, si es que existe alguna.

Cuando alguna regla del archivo sea satisfecha, se ejecutarán sus respectivas instrucciones.

Para mayor información respecto a los Makefiles pueden visitar el siguiente tutorial en Internet:

<http://www.eng.hawaii.edu/Tutor/Make/>.

## 2. Introducción a nSystem

De acuerdo a la documentación de nSystem, este es un:

*...sistema de procesos livianos para Unix con fines pedagógicos. El sistema consiste de unos cientos de líneas de código en C (la mitad son comentarios) que implementan:*

- *Creación y destrucción de procesos livianos (tareas). Estos procesos comparten un mismo espacio de direcciones (están dentro de un solo proceso UNIX).*
- *Paso de Mensajes para la sincronización de tareas.*
- *Entrada/Salida no bloqueante para el proceso UNIX.*
- *Un scheduler muy simple. Permite implementar administración preemptive y non-preemptive.*

En términos prácticos, nSystem es una librería que permite administrar threads dentro de un solo proceso pesado Unix. Para poder utilizar nSystem se deben llevar a cabo los siguientes pasos:

1. Bajar nSystem desde Ucourses <http://www.dcc.uchile.cl/~lmateu/CC41B/> El cual estará disponible a más tardar con el enunciado de la tarea 1.
2. Descomprimirlo y configurar la variable de ambiente `$NSYSTEM` con el `PATH` de nSystem. Por ejemplo:
  - En `tcsh`: `setenv NSYSTEM /home/micarpeta/nSystem`
  - En `bash`: `export NSYSTEM=/home/micarpeta/nSystem`

3. Compilar nSystem

```
cd $NSYSTEM/src
make
```

para Linux. Para utilizarlo en Solaris debe comentar y descomentar las líneas respectivas.

4. Compilar el programa que utiliza nSystem de la siguiente manera:

```
gcc -I$NSYSTEM/include/ pruebat1.c tarea1.c $NSYSTEM/lib/libnSys.a -o prueba
```

es el archivo de prueba para la tarea. Como observación y a modo que preparen los Makefiles de entrega para las tareas, lo anterior es equivalente a:

```
gcc -c -I$NSYSTEM/include/ tarea1.c
gcc -c -I$NSYSTEM/include/ pruebat1.c
gcc -I$NSYSTEM/include/ pruebat1.o tarea1.o $NSYSTEM/lib/libnSys.a -o prueba
```