

Make

Esp. Ing. José María Sola, profesor

Revisión 1.0.0

2022-12-04

Tabla de contenidos

1. Introducción a <i>make</i> : Ejercicios de Repaso	1
1.1. ¿Qué es <i>make</i> ?	1
1.2. ¿Qué facilita <i>make</i> ?	1
1.3. ¿Qué es un <i>makefile</i> ?	1
1.4. ¿Qué partes tiene una regla de <i>Makefile</i> ?	2
1.5. Ejercicio #2	2
1.6. Ejercicio #3	2
1.7. Investigación	2
2. Referencia	3
3. Changelog	5

1

Introducción a *make*: Ejercicios de Repaso

1.1. ¿Qué es *make*?

Es una herramienta que determina automáticamente que partes de un programa grande necesitan recompilarse, y emite los comandos para hacerlo. Efectivamente *hace* (i.e., *makes*) el programa.

1.2. ¿Qué facilita *make*?

La actualización automática de archivos desde otros archivos, que se disparan cuando los segundos se modifican. Automatiza el proceso de *building* (i.e., "*buildeo*" o traducción) de un programa grande. Permite actualizar solo lo que cambió, sin necesidad de recompilar todos los archivos fuente que componen el programa.

1.3. ¿Qué es un *makefile*?

Es un archivo que con una notación especial que describe las relaciones entre los archivos de tu programa y provee los comandos para actualizar cada archivo.

En un programa, típicamente, un ejecutable es actualizado desde archivos objetos, que su vez son *hechos* (i.e., *made*) mediante la compilación de archivos fuentes.

El programa *make* utiliza las definiciones en el *makefile* y las fechas de modificación de los archivos para decidir qué archivos necesitan actualizarse. Para cada uno de esos archivos, emite las recetas definidas en el *makefile*.

1.4. ¿Qué partes tiene una regla de *Makefile*?

```
objetivo ... : prerequisites ...  
    receta  
    ...  
    ...
```

- Ejercicio #1: Describa cada una de las partes.

1.5. Ejercicio #2

Suponga el siguiente *makefile*

```
A.o: A.c A.h  
    cc A.c -o A.o
```

1. ¿Qué ocurre si corremos `make`?
2. ¿Qué ocurre si corremos `make` de nuevo?
3. ¿Qué ocurre si hacemos una modificación en la interfaz (`A.h`) y ejecutamos `make` de nuevo?

1.6. Ejercicio #3

Escriba un *Makefile* para un ejecutable que depende de un fuente y dos módulos.

Dé nombres significativos a los archivos, de tal forma que representen la solución a un posible problema real.

1.7. Investigación

Busque proyectos reconocidos que utilicen *Makefiles*. Por ejemplo, ¿dónde está el *Makefile* para *Linux*?

2

Referencia

- GNU Make: A Program for Directing Compilation. <https://www.gnu.org/software/make/manual/make.pdf>

3

Changelog

1.0.0+2022-12-04

- Versión inicial.