



 Grado en Ingeniería Informática

Seguridad y Riesgos en Sistemas de Información

Práctica 1 - Herramientas

Laboratorio

Contenidos

Introducción	3
Herramientas	4
Tmux	4
Conceptos Básicos	4
Primeros Pasos	5
Gestión de Sesiones	6
Vim	7
Bibliografía	8

Introducción

En los grados de ingeniería informática se trabajan multitud de materias, desde sistemas operativos e ingeniería del software a sistemas inteligentes y ubicuos. Pero hay un aspecto que raramente se trata, dejando a los estudiantes la elección de las herramientas para ser más eficientes [6].

En esta primera parte de la práctica se introducen algunas de las herramientas con las que 1) el administrador de sistemas se va a encontrar en más de una ocasión, *vim* [7], y 2) le permitirán ser más eficiente en la realización de sus tareas, *tmux* [2].

Herramientas

Cuando se trabaja como administrador de sistemas, y especialmente si algo “malo” ha sucedido, es más que probable que el trabajo se realice mediante un terminal (sobre la máquina física, mediante una conexión ssh, ...). Existen multitud de herramientas que facilitan la gestión del sistema en estos escenarios, entre las que destacamos la edición de ficheros y el trabajo en múltiples terminales.

A la hora de editar texto, más allá de las capacidades de edición de utilidades básicas como *grep*, *sed* o *find*, por enumerar algunos de ellos, hay situaciones donde se precisa de una edición más compleja. *Vim* es un editor de texto que se encuentra disponible en prácticamente cualquier entorno Linux. Su ubicuidad, entre otros factores [5] y [9], hacen de Vim una herramienta sobre la que es deseable un conocimiento básico, más allá de la complejidad que presenta.

Cuando se trabaja en la terminal generalmente es necesario, o por lo menos facilita el trabajo, disponer de múltiples instancias. Para ello podemos utilizar *Tmux*, que nos permitirá trabajar con múltiples sesiones en una sola sesión.

Tmux

Tmux es un multiplexor de la terminal que permite la creación de múltiples instancias desde una sola pantalla, así como la conexión/desconexión de estas con ejecución en segundo plano. Su principal utilidad es la de tener múltiples instancias o sesiones en un sólo terminal. En entornos donde tenemos un acceso limitado a los recursos, o para permitir realizar varias operaciones de forma simultánea o incluso en segundo plano, es una herramienta indispensable. Se puede instalar utilizando *apt*:

```
$ sudo apt install tmux
```

Algunos de los usos principales de *tmux* serían¹:

- Proteger la ejecución de programas en un servidor remoto de pérdidas de conexión
- Permitir cerrar *tmux*, pero quedando la ejecución en segundo plano pudiéndola recuperar en cualquier momento
- Trabajar con múltiples programas y *shells* en una sola terminal. Esta función tiene similitudes con los gestores de ventanas

Conceptos Básicos

Cada **panel** contiene un terminal donde se ejecuta un programa. Estos paneles se encuentran distribuidos en una **ventana**. La ventana ocupa habitualmente todo el espacio del terminal donde se ejecuta *tmux*. Sobre este punto de partida podemos definir los siguientes conceptos:

¹Más información en <https://github.com/tmux/tmux/wiki/Getting-Started>

- Cada ventana tiene un **nombre** y un panel activo
- Las ventanas están enlazadas a una o más **sesiones**
- Cada **sesión** tiene una lista de ventanas, cada una de ellas con un **índice**
- Una de las ventanas en la sesión es la ventana **activa**
- Cada sesión se encuentra conectada a uno o más **clientes**, o se encuentra desconectada, sin clientes
- Cada cliente se encuentra conectado a una sesión

Término	Descripción
Cliente	Conecta una sesión <i>tmux</i> desde un terminal externo
Sesión	Agrupar una o más ventanas
Ventana	Agrupar uno o más paneles, enlazados a una o más sesiones
Panel	Contiene un terminal ejecutando programas, aparece en una ventana
Panel activo	El panel en la ventana actual al cual se envía la entrada del usuario; uno por ventana
Ventana actual	La ventana en la sesión actual a la que se envía la entrada del usuario; uno por sesión
Última ventana	La ventana actual anterior
Nombre de sesión	El nombre de la sesión. Por defecto se numera comenzando por cero
Listado de ventanas	El listado de las ventanas en una sesión ordenada por nombre
Nombre de ventana	El nombre de una ventana. Por defecto toma el nombre del programa del panel activo
Índice de ventana	El número de una ventana en la lista de ventanas de la sesión
Disposición de ventana	El tamaño y posición de los paneles en una ventana

Primeros Pasos

Para crear una nueva sesión se ejecuta *tmux* en el terminal. Se utiliza el argumento *new*, aunque si no se utiliza ningún argumento tiene el mismo efecto:

```
$ tmux new
```

Si se utiliza el argumento *-s* se puede especificar el nombre de la sesión, por ejemplo “desarrollo”:

```
$ tmux new -s desarrollo
```

Una vez conectados a una sesión tendremos a nuestra disposición un panel con una ventana. La gestión de paneles y ventanas se realiza mediante atajos de

teclado. Estos atajos de teclado utilizan un prefijo que consiste en una combinación de teclas. Por defecto, el prefijo es *ctrl+b*.

Para una gestión básica de las ventanas podemos utilizar los siguientes atajos:

Combinación	Acción
ctrl+b c	Crea una ventana
ctrl+b ,	Renombra la ventana actual
ctrl+b &	Cierra la ventana actual
ctrl+b w	Lista ventanas
ctrl+b p	Ventana actual
ctrl+b n	Ventana siguiente
ctrl+b 0...9	Selecciona ventana por número

Para una gestión básica de los paneles podemos utilizar los siguientes atajos:

Combinación	Acción
ctrl+b %	Divide panel con distribución horizontal
ctrl+b ”	Divide panel con distribución vertical
ctrl+b {	Mueve el panel actual a la izquierda
ctrl+b }	Mueve el panel actual a la derecha
ctrl+b flechas	Cambia el panel según la dirección
ctrl+b z	Maximiza un panel
ctrl+b !	Convierte un panel en una ventana

Para cambiar el tamaño de los paneles se ejecuta *ctrl+b+[flecha]*, repitiendo la pulsación en flecha para aumentar/decrementar sucesivamente su tamaño.

Para mostrar todos los atajos de teclado utilizar la combinación *ctrl+b ?*.

Gestión de Sesiones

Al estar conectados a una sesión de *tmux* podemos salir de ella cerrando todos los paneles de todas las ventanas. Esto hará que la sesión se cierre. Pero podemos salir de la sesión sin cerrar, bien de forma voluntaria o por que se produzca una desconexión con el servidor.

Estando conectados podemos desconectarnos de la sesión (sin cerrarla) utilizando el atajo *ctrl+b d*. Al hacerlo, saldremos de la sesión y se nos informará de que nos hemos desconectado de la sesión correspondiente. Por ejemplo, a continuación se muestra el mensaje que se recibe al desconectarse de la sesión “0”:

```
$ tmux
[detached (from session 0)]
$
```

Si queremos ver las sesiones activas utilizaremos el argumento *ls*. En el siguiente ejemplo se muestran cuatro sesiones con los nombre “0”, “1”, “config” y “desarrollo”

```
$ tmux ls
0: 3 windows (created Fri Dec 23 17:04:41 2023) (attached)
1: 2 windows (created Fri Dec 23 17:21:53 2023)
config: 4 windows (created Fri Dec 23 17:24:30 2023)
desarrollo: 1 windows (created Fri Dec 23 17:24:15 2023)
```

Para conectar con una sesión, en lugar de crear una nueva, se utiliza el argumento *attach -t* seguido del nombre de la sesión. Para conectar con la sesión de nombre *config*, según el ejemplo anterior, se utilizaría el siguiente comando:

```
$ tmux attach -t config
```

También es posible eliminar sesiones desde la línea de comandos. En este caso utilizaremos el parámetro *kill-session -t*. En el siguiente ejemplo se elimina la sesión de nombre *1*:

```
$ tmux kill-session -t 1
```

Vim

Vim es un editor de texto que permite una edición avanzada de archivos de texto con multitud de funcionalidades tales como registros, macros, *plugins* y capacidad de personalización para múltiples lenguajes o entornos. Se puede instalar utilizando *apt*:

```
$ sudo apt install vim
```

También se puede instalar una versión más grande que proporciona algunas funcionalidades deseables, como el acceso al *clipboard* del sistema para copia/pegar texto entre *Vim* y el resto de aplicaciones. Para ello hay que instalar el paquete *vim-gtk*:

```
$ sudo apt install vim-gtk
```

Existen algunos recursos para aprender Vim. Lo idóneo es comenzar con *vimtutor*, utilidad que se instala al instalar el editor *Vim*. *Vimtutor* se compone de 7 lecciones que permiten aprender algunas de las funciones básicas de *Vim*.

```
$ vimtutor
```

También se puede aprender a utilizar los fundamentos de vim a través del juego online *vim-adventures*² o de multitud de recursos online, como por ejemplo los ofrecidos por el *Massachusetts Institute of Technology* [5] como parte de la serie

²VIM Adventures: <https://vim-adventures.com/>

The Missing Semester of Your CS Education [6]. Para conseguir un mayor dominio en el manejo de esta herramienta, o para realizar una introducción más completa, uno de los mejores, si no el mejor, recurso al respecto es el publicado por Neil [8], *Practical Vim*.

Finalmente, cabe destacar que existen multitud de aplicaciones que utilizan modos compatibles con la forma en que se interactúa con *vim*. Algunos ejemplos son gestores de ficheros en la línea de comandos, como *vifm* [3] o *ranger* [10], o herramientas de edición de código y texto como *IntelliJ* [4] u *Overleaf* [1].

Bibliografía

- [1] 2024. Overleaf. <https://www.overleaf.com>.
- [2] 2023. tmux. <https://github.com/tmux/tmux>.
- [3] 2024. Vifm file manager. <https://www.vifm.info/>.
- [4] JetBrains 2024. IntelliJ. <https://www.jetbrains.com/es-es/idea/>.
- [5] Massachusetts Institute of Technology 2020. Lecture 3: Editors (vim) (2020). <https://missing.csail.mit.edu/2020/editors/>.
- [6] Massachusetts Institute of Technology 2020. The Missing Semester of Your CS Education. <https://missing.csail.mit.edu/>.
- [7] Moolenaar, Bram 2023. Vim the editor. <https://www.vim.org/>.
- [8] Neil, D. 2015. *Practical Vim: Edit Text at the Speed of Thought*. Pragmatic Bookshelf.
- [9] The Linux Foundation 2022. Classic SysAdmin: Vim 101: A Beginner's Guide to Vim. <https://www.linuxfoundation.org/blog/blog/classic-sysadmin-vim-101-a-beginners-guide-to-vim>.
- [10] toonn 2024. Ranger file manager. <https://github.com/ranger/ranger>.