



Universidad Nacional  
Autónoma de México

DGTIC



**supercómputo**  
Plan de Becarios



# ALTA DEL SISTEMA (PARTE 2)

M. I. Lourdes Yolanda Flores Salgado  
yoli@unam.mx

DEPARTAMENTO DE SUPERCÓMPUTO  
DGTIC-UNAM



# UPSTART

- Sistema de inicio basado en eventos.
  - Ubuntu
  - Google Chrome OS
  - Linux RH 6 (LRHE, CentOS, Fedora)
  - Debian\*
  - OpenSUSE\*
- Sustituye al tradicional proceso init añadiendo nuevas características.
- Entre las mejoras destacan el arranque de servicios en paralelo, guiado por eventos, servicios que levantan de forma automática si mueren, etc.



## UPSTART (INIT)

### Linux Red Hat

- El proceso init de upstart, inicia, supervisa y detiene los procesos del sistema definidos en el directorio `/etc/init`
- Sólo utiliza `/etc/inittab` para conocer el nivel de default del sistema.
- Los servicios pueden ser iniciados o detenidos en respuesta a cualquier evento del sistema, sin importar que evento sea.



## UPSTART (INIT)

### Linux Red Hat

- El evento primario, generado por el demonio init, se llama “startup”.
- Startup inicia la cadena de eventos que darán de alta el sistema.





# UPSTART (INIT) ¿CÓMO FUNCIONA?

## Linux Red Hat

- Las tareas o servicios manejados por upstart son conocidos como “Jobs”.
- Los eventos disparan los jobs (definidos en `/etc/init`).
- Comúnmente, un job es un simple script en shell.
- Los jobs más complejos pueden incluir código para ejecutar antes de que el job sea iniciado o después de que finalizó.



## EJEMPLO DE UN JOB

upstart

description “set system hostname”

start on startup

task

exec hostname -b -F /etc/hostname



# UPSTART (INIT)

## Linux Red Hat

### Compatibilidad con System V

- Por sí mismo, upstart no maneja niveles de sistema.
- Los niveles se manejan por medio de la implementación del evento *runlevel*.
- `/etc/init/rc.conf` Es el job que se inicia cuando se presenta el evento *runlevel* y está encargado de ejecutar el archivo `/etc/rc.d/rc`



## SYSTEMD

- Gestor del sistema y servicios que reemplaza a init de SV.
- Es el primer proceso del sistema PID=1 e inicia al resto de los procesos del sistema.





# SYSTEMD CARACTERÍSTICAS

- Capacidades de paralalización.
- Inicio de demonios bajo demanda.
- Gestión de dependencias.
- Monitoreo de procesos.
- Compatible con SysV init



## SYSTEMD

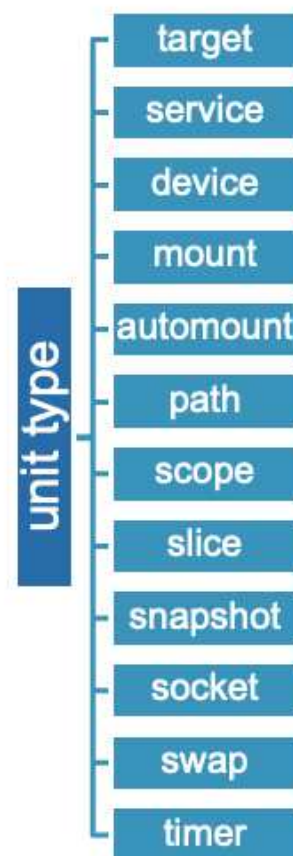
- Introduce el concepto de unidades (units) que son diferentes tipos de objetos del sistema.
- Las unidades más comunes son:
  - target Utilizada para la agrupación lógica de unidades.
  - service Demonios que pueden ser iniciados, detenidos, etc.

# UNIDADES SYSTEMD

nombre.tipo

Ejemplo:

apache.service



## UNIDADES SYSTEMD

`/usr/lib/systemd/system`

Instaladas con el S.O.

`/run/systemd/system`

Creadas en tiempo de ejecución

`/etc/systemd/system`

Creadas por el sysadmin





## SYSTEMCTL

- Comando que se utiliza para administrar las unidades de systemd.
- Listar unidades



# SYSTEMCTL

- Listado de unidades activas:

```
$ systemctl
```

```
$ systemctl list-units
```

- Listado de unidades que han tenido problemas:

```
$ systemctl --failed
```

- Se puede ver un listado de las unidades instaladas con:

```
$ systemctl list-unit-files
```



# SYSTEMCTL

- Activa una unidad de inmediato:

```
# systemctl start unidad
```

- Desactiva una unidad de inmediato:

```
# systemctl stop unidad
```

- Reinicia la unidad:

```
# systemctl restart unidad
```



# SYSTEMCTL

- Hace que una unidad recargue su configuración:

```
# systemctl reload unidad
```

- Muestra el estado de una unidad, incluso si se está ejecutando o no:

```
$ systemctl status unidad
```

- Comprueba si la unidad ya está habilitada o no:

```
$ systemctl is-enabled unidad
```

- Activa el inicio automático en el arranque:

```
# systemctl enable unidad
```





## SYSTEMCTL

- Desactiva el inicio automático durante el arranque:

```
# systemctl disable unidad
```

- Muestra la página del manual asociada con una unidad (esto tiene que ser apoyado por el archivo .unit):

```
$ systemctl help unidad
```



# SYSTEMCTL

- Apagado y reinicio del sistema:

```
$ systemctl reboot
```

- Apagado del sistema:

```
$ systemctl poweroff
```

- Suspensión del sistema:

```
$ systemctl suspend
```

- Poner el sistema en hibernación:

```
$ systemctl hibernate
```



# UNIDADES SYSTEMD

- Estructura básica:

[Unit]

Description=Descripción de la unidad

After=Define el orden de arranque de las unidades, la unidad sólo arranca después de que las unidades aquí listadas hayan arrancado

Requires=Dependencias (las configura si son de otra unidad y las arranca, si hay un fallo en estas la unidad no arranca)

[Service]

Type=Configura el proceso de arranque de la unidad, por default es “simple”, otras opciones son “forking”, “oneshot”

ExecStart=Ruta al binario y argumentos

ExecStop=Ruta al script o comando ejecutado cuando la unidad es detenida (opcional)

[Install]

WantedBy=Target que va a usar la unidad

# EJEMPLO POSTFIX.SERVICE

## [Unit]

Description=Postfix Mail Transport Agent  
After=syslog.target network.target  
Conflicts=sendmail.service exim.service

## [Service]

Type=forking  
PIDFile=/var/spool/postfix/pid/master.pid  
EnvironmentFile=-/etc/sysconfig/network  
ExecStartPre=-/usr/libexec/postfix/aliasesdb  
ExecStartPre=-/usr/libexec/postfix/chroot-update  
ExecStart=/usr/sbin/postfix start  
ExecReload=/usr/sbin/postfix reload  
ExecStop=/usr/sbin/postfix stop

## [Install]

WantedBy=multi-user.target





# Systemd (demonios)

## systemd Utilities

systemctl journalctl notify analyze cglsg cgtop loginctl nspawn

## systemd Daemons

systemd  
journalld networkd  
logind user session

## systemd Targets

bootmode basic multi-user graphical user-session  
shutdown reboot dbus telephony display service  
dlog logind user-session tizen service

## systemd Core

manager unit login namespace log  
systemd service timer mount target multiseat inhibit  
snapshot path socket swap session pam cgroup dbus

## systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

## Linux Kernel

cgroups autofs kdbus