

Pràctica4: grub

Curs 2019 - 2020

ASIX M01-ISO Implantació de sistemes operatius
UF3 Arrancada

Pràctica a fer

Usarem el recurs creat en la pràctica anterior al vostre [gitlab](#) anomenat **asix-m01** d'accés públic. Deseu els exercicis a fer d'aquesta pràctica en aquest recurs. Envieu un email al professor on consti clarament el vostre nom i cognoms i un enllaç al recurs de gitlab on hi ha les vostres solucions *informant que ja heu acabat aquesta quarta pràctica*. Teniu fins al 07/05/2020. En el gitlab genereu una carpeta anomenada **practica4**.

Repositori del professor

Recordeu que a classe hem estat treballant amb el repositori de gitlab on hi ha penjat el material que hem estat fent a classe gitlab de edtasixm01 recurs [Scripts-2019-2020](#).

Recordeu també que el material de l'assignatura està a la web de Google Sites [ASIX-M01](#) i que ara estem treballant la part [d'arrancada del sistema](#).

Una mica de descripció i repàs del què farem

Seguint el dossier pdf que s'adjunta en aquest document anomenat **[HowTo-ASIX-Arrancada-Systemd](#)** practicareu i aprendreu els següents conceptes:

- ☐ Identificar l'arquitectura de systemd
- ☐ Indentificar el procediment global d'arrancada
- ☐ Els targets
- ☐ Els services
- ☐ Systemctl per activar/desactivar i enable/disable
- ☐ El directori Wants
- ☐ Diferència entre /etc/... i /usr/lib/...

Mètode de treball

Per practicar us recomano utilitzar una màquina virtual senzilla. Evidentment podeu fer-ho al vostre host però essent curosos amb les accions que realitzeu, errors tipogràfics poden fer que el sistema no arrenqui.

La meva recomanació principal és usar una de les màquines lleugeres de Fedora 27 Cloud, per exemple la versió de qcow2 que és molt lleugera. Ocupa més la imatge a descarregar que la raw.xz però un cop descarregada la raw creix fins els 4GB i la qcow2 no.

Exercicis a fer:

- Llegiu tot el document HowTo-ASIX-Arrancada-Systemd i aneu fent els exercicis que hi apareixen.
- Cal desar en un document al Git els exercicis i les solucions.
- Hi ha una pràctica final de generar targets propis amb serveis personalitzats. Cal incloure també la solució d'aquesta pràctica.

Extracció dels exercicis del Howto:

Targets d'arrencada

1. En tots els casos verificar el target actual. Llisteu els processos i els serveis.
2. Configurar el sistema establint default.target a mode multi-user.target.
3. Manualment canviar a graphical.target amb isolate.
4. Reiniciar el sistema i al grub establir l'opció de iniciar en mode emergency.target.
5. Reiniciar i indicar al grub l'opció del kernel per iniciar a rescue.target.
6. Canviar de target amb l'ordre isolate activant multi-user.target.
7. Restablir per defecte graphical.target i reiniciar el sistema.
8. Amb isolate indicar que es vol accedir al target poweroff.target.
9. Iniciar el sistema en mode emergency.target. Llistar els processos, l'arbre de processos, els units i les dependències. Cal indicar el password de root? Hi ha múltiples sessions de consola?
10. Iniciar el sistema en mode init=/bin/bash. Llistar els processos, l'arbre de processos, els units i les dependències. Cal indicar el password de root? Hi ha múltiples sessions de consola?

Gestió bàsica de serveis

11. Instal·leu els serveis httpd, gpm, xinetd i ssh.
12. Activeu els serveis httpd i ssh i configureu-los disabled. Verifiqueu que estan actius. Reiniciar el sistema i observar que no estan Iniciar el sistema.
13. Amb els serveis inactius configurar-los (httpd i sshd) com a enabled. Verificar que estan enables però inactius. Reiniciar el sistema i observar que estan actius i enabled.
14. Fer mask del servei httpd i observar que no es pot fer enable. Fer unmask.

Serveis i Units

15. Identificar el fitxer .service del servei xinetd. Identificar l'executable.
16. Configurar manualment amb symlinks el enable del servei xinetd i verificar-ho.
17. Eliminar el enable del servei xinetd manualment amb el symlink. Verificar-ho.

Arquitectura de systemd

18. Instal·la el paquet xinetd. Llista els seus components.
19. Identifica el fitxer de servei de xinetd i llista'l. Identifica l'executable del servei i a quin target s'instal·la.
20. Llista el directori on hi ha els fitxers de les unitats de servei de multi-user.target. Identifica el fitxer corresponent al servei xinetd.
21. Fes enable del servei xinetd. S'ha creat un link. Llista el directori origen i el directori destí del link. On s'ha creat el link? Perquè?
22. Llista les dependències de multi-user.target: totes; només els targets; només els serveis.
23. Llista les dependències de xinetd.service.
24. Fes l'ordre systemctl show del servei xinetd i identifica els elements: After i WantedBy.
25. Es pot iniciar sessió en el sysinit.target? Explica el perquè documentant-ho.

Arrencada de targets diferents

26. Mostra les dependències de rescue.target: totes; targets; serveis.
27. Mostra les dependències de emergency.target: totes; targets; serveis.
28. Estableix com a target per defecte reboot.target i reinicia la màquina.
29. Punyeta! Arregla-ho!

Exercici pràctic:

- Fer un servei [myhello.service](#) on generem un missatge de benvinguda. Crear el .service de tipus [oneshot](#) que significa que s'executa un cop i finalitza. Observeu que s'ha comentat la línia [StandardOutput](#) perquè el echo no vagi al jorural sinó a la pantalla. El servei s'instal·larà en un target personalitzat anomenat [mytarget.target](#).

```
[Unit]
Description=El meu primer servei

[Service]
Type=oneshot
ExecStart=/usr/bin/echo "bye bye champions"
RemainAfterExit=true
ExecStop=/usr/bin/echo "liverpool"
#StandardOutput=journal

[Install]
WantedBy=mytarget.target
```

- Fer un servei [myweb.service](#) on creen un servei web basat en el servei de httpd. Cal instal·lar prèviament el paquet httpd. Crear-lo de tipus [fork](#) engegant l'executable de httpd. El servei s'instal·larà en un target personalitzat anomenat [mytarget.target](#).

```
[Unit]
Description=My web server

[Service]
Type=forking
```

```
ExecStart=/usr/sbin/httpd
ExecStop=/usr/bin/echo "aturant web server myweb"
ExecStop=/usr/bin/pkill httpd
StandardOutput=journal
```

```
[Install]
WantedBy=mytarget.target
```

- Fer un target d'arrencada personalitzat anomenat [mytarget.target](#) basat en [rescue.target](#) que tingui el servei [gpm](#), [myhello](#) i el [myweb](#).

```
[Unit]
Description=El meu mode de rescat mytarget
Documentation=man:systemd.special(7)
Requires=sysinit.target rescue.service
After=sysinit.target rescue.service
AllowIsolate=yes
```

```
[Install]
Alias=rescat.target
```

- Pensar a fer el [systemctl daemon-reload](#). Sempre que es modifiquen els units (el contingut dels fitxers dels units) cal fer [systemctl daemon-reload](#) perquè [systemd](#) recarregi la configuració de les units.
- Fer l'enable dels serveis [myhello](#) i [myweb](#). Verificar amb [list dependencies](#) les dependències de [mytarget.target](#). Llistar el seu directori [.wants](#).
- Reboot i iniciar el sistema indicant en el GRUB el target [mytarget.target](#). Verificar amb [is-active](#) i [list-dependencies](#) que està actiu i què no està actiu.
- Repetir-ho però en lloc de usar [myweb.service](#) (apagar i disable de [myweb](#)) usar [myssh.service](#) o [myhttp.service](#), tunejant els serveis originals [http](#) i [sshd](#). Recordeu de fer el [daemon-reload](#). Observar que si es requereix de xarxa per funcionar [systemd](#) 'estira' de ella.