

Un esempio pratico su come **combinare** le potenzialità del **Cronjob Linux** con il **Versioning Service Git**

1) Creiamo anzitutto la cartella in cui attivare il servizio di versioning Git; nel mio esempio, mi trovavo nella directory esterna **WorkShop**, ed ho chiamato la directory dove attivare Git **GitE**; il path dei file su cui operare il versioning è pertanto **WorkShop/GitE**.

Successivamente **attiviamo il servizio Git** e prepariamo i **due files** (**FileData1.txt** e **FileData2.txt**) su cui operare inizializzandoli e creando una (o due) **loro prima commit** di inzializzazione.

Tutte queste operazioni (partendo dalla super-directory **WorkShop**) saranno attuate digitando i seguenti comandi bash:

```
mkdir GitE  
cd GitE
```

```
git init
```

```
git config user.name "Me Stesso"  
git config user.email "miamail@mailservice.com"  
alias mostra="git log --all --graph --decorate --oneline"
```

```
echo "Start" >> FileData1.txt  
echo "Start" >> FileData2.txt
```

```
git add *.txt  
git commit *.txt -m "Commit Start"
```

2) Creiamo un file esecutore di tipo shell e di nome **executer.sh**, sempre all'interno della directory **GitE**; questo file dovrà contenere le seguenti righe di codice:

```
current="`date +%Y-%m-%d %H:%M:%S`"  
msg="Updated: $current"
```

```
git -C /root/WorkShop/GitE/ branch BR1  
git -C /root/WorkShop/GitE/ branch BR2  
git -C /root/WorkShop/GitE/ checkout BR1  
git -C /root/WorkShop/GitE/ branch -d Main
```

```
date >> /root/WorkShop/GitE/FileData1.txt  
git -C /root/WorkShop/GitE/ commit FileData1.txt -m "BR1 - $msg"
```

```
git -C /root/WorkShop/GitE/ checkout BR2  
date >> /root/WorkShop/GitE/FileData2.txt  
git -C /root/WorkShop/GitE/ commit FileData2.txt -m "BR2 - $msg"
```

```
git -C /root/WorkShop/GitE/ merge BR1 -m "Fusion"
```

```
git -C /root/WorkShop/GitE/ branch Main  
git -C /root/WorkShop/GitE/ checkout Main  
git -C /root/WorkShop/GitE/ branch -d BR1  
git -C /root/WorkShop/GitE/ branch -d BR2
```

3) Rendiamo l'esecutore eseguibile digitando il comando bash seguente:

```
chmod +x ./executer.sh
```

4) Apriamo il file di registro dei Cronjobs con il comando bash seguente:

```
crontab -e
```

ed inseriamo al suo interno la seguente riga di comando:

```
*/1 * * * * /root/WorkShop/GitE/executer.sh
```

5) Lasciamo scorrere il tempo e, quando avremo tempo, **digitiamo** il comando bash che attiva il *git log* per conoscere lo stato delle commits:

```
mostra
```

che è l'*alias* creato al punto (1), oppure digitiamo direttamente il comando originario:

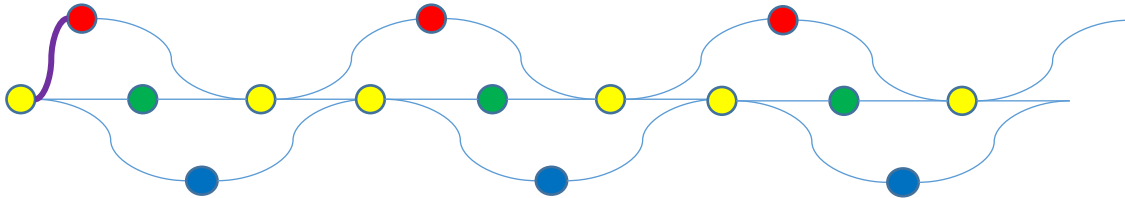
`git log --all --graph --decorate --oneline`

Il risultato sarà la ***catena delle commits***, come mostrato di seguito:

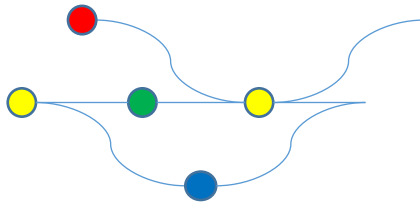
```
* 24f1d72 (HEAD -> Main) Fusion
| \
| * 5732bf8 BR1 - Updated: 2023-11-10 20:00:01
* | 14ed601 BR2 - Updated: 2023-11-10 20:00:01
| /
* 4e990f6 Fusion
| \
| * 4277ef1 BR1 - Updated: 2023-11-10 19:59:01
* | 486377c BR2 - Updated: 2023-11-10 19:59:01
| /
* c77cd58 Fusion
| \
| * b9b72cd BR1 - Updated: 2023-11-10 19:58:01
* | e677cf5 BR2 - Updated: 2023-11-10 19:58:01
| /
* bbf7382 Fusion
| \
| * c1ab0a7 BR1 - Updated: 2023-11-10 19:57:01
* | 102a86d BR2 - Updated: 2023-11-10 19:57:01
| /
* c346d80 Commit Start
* 32a0d0b commit BR1
* 9c7bc25 commit BR1
:|
```

Come effettuare una catena di commits

Qui mostriamo un codice-script da eseguire in una shell Linux o in Git Bash per creare una catena di commits che segua uno scham come quello riportato di seguito:



L'unità ripetuta di questa esecuzione periodica è quella mostrata di sotto:



Il codice-script utilizzato per realizzarla e l'output di ***git log*** sono i seguenti:

```

declare -i Nturns=5

git init
git config user.name "Me Stesso"
git config user.email "miamail@mailservice.com"
echo "Start" >> FileData1.txt
echo "Start" >> FileData2.txt
echo "Start" >> FileData3.txt
git add *.txt
git commit *.txt -m "Commit Start"

git branch BR2
git branch BR1
git checkout BR2
git branch -d Main

for (( c=1; c<=$Nturns; c++ )); do

git checkout BR1
current=""date +%Y-%m-%d %H:%M:%S""
msg="Updated: $current"
date >> FileData1.txt
git commit FileData1.txt -m "BR1 - $msg"

git checkout BR2
git branch BR3

# git checkout BR2
current=""date +%Y-%m-%d %H:%M:%S""
msg="Updated: $current"
date >> FileData2.txt
git commit FileData2.txt -m "BR2 - $msg"

git checkout BR3
current=""date +%Y-%m-%d %H:%M:%S""
msg="Updated: $current"
date >> FileData3.txt
git commit FileData3.txt -m "BR3 - $msg"

git checkout BR2
current=""date +%Y-%m-%d %H:%M:%S""
msg="Updated: $current"
git merge BR1 -m "Fusion BR1-BR2 - $msg"
git branch -d BR1
git branch BR1

current=""date +%Y-%m-%d %H:%M:%S""
msg="Updated: $current"
git merge BR3 -m "Fusion BR3-BR2 - $msg"
git branch -d BR3

done

git log --all --graph --decorate --oneline

```

```

* bd66013 (HEAD -> BR2) Fusion BR3-BR2 - Updated: 2023-11-12 10:04:21
|
| * a780447 BR3 - Updated: 2023-11-12 10:04:18
| * a06dc61 (BR1) Fusion BR1-BR2 - Updated: 2023-11-12 10:04:20
|
| * 6e2ad40 BR1 - Updated: 2023-11-12 10:04:16
| * 2b525b0 BR2 - Updated: 2023-11-12 10:04:17
|
|
| * 92fb4c9 Fusion BR3-BR2 - Updated: 2023-11-12 10:04:15
|
|
| * b253c3d BR3 - Updated: 2023-11-12 10:04:12
| * a1c3d7f Fusion BR1-BR2 - Updated: 2023-11-12 10:04:13
|
| * a6f8768 BR1 - Updated: 2023-11-12 10:04:10
| * 7fefdc68 BR2 - Updated: 2023-11-12 10:04:11
|
|
| * 9e5f4e2 Fusion BR3-BR2 - Updated: 2023-11-12 10:04:08
|
|
| * 2249763 BR3 - Updated: 2023-11-12 10:04:05
| * 400b9ab Fusion BR1-BR2 - Updated: 2023-11-12 10:04:07
|
| * c94e9e4 BR1 - Updated: 2023-11-12 10:04:03
| * 6434c9d BR2 - Updated: 2023-11-12 10:04:04
|
|
| * 33a597f Fusion BR3-BR2 - Updated: 2023-11-12 10:04:01
|
|
| * dd3e237 BR3 - Updated: 2023-11-12 10:03:59
| * 0a5aa86 Fusion BR1-BR2 - Updated: 2023-11-12 10:04:00
|
| * 45210e8 BR1 - Updated: 2023-11-12 10:03:56
| * bfa01c6 BR2 - Updated: 2023-11-12 10:03:57
|
|
| * 32c56b8 Fusion BR3-BR2 - Updated: 2023-11-12 10:03:54
|
|
| * 48eb772 BR3 - Updated: 2023-11-12 10:03:51
| * 243b088 Fusion BR1-BR2 - Updated: 2023-11-12 10:03:52
|
| * e603d1a BR1 - Updated: 2023-11-12 10:03:49
| * f2b3f70 BR2 - Updated: 2023-11-12 10:03:50
|
| * 8a5984d Commit Start

```