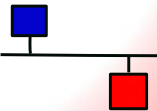


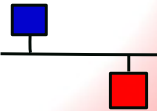
La Shell Lavora Per Noi - 3

Come accelerare compiti comuni
Utilizzando la Shell, la sua programmazione ed i
comandi di Linux
Soluzione a tre schermi – uno clonato l'altro a lato



La situazione

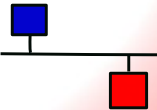
- Devo fare una dimostrazione di un programma, ad esempio Freecad
- Devo proiettare lo schermo principale su cui lavoro, quindi lo voglio clonare sul proiettore (secondo schermo)
- Mi servono degli appunti dai quali copiare i date, che però non voglio mostrare e quindi voglio utilizzare un terzo schermo per quelli



La gestione del parametro

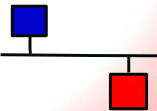
- Dobbiamo consentire di mettere lo schermo terziario **a destra**, **sopra** o **a sinistra** del primario.
- Per default sarà a destra:

```
POSITION='right-of'
if [ -n "$1" ]
then
    if [ $1 = '-l' ]
    then
        POSITION='left-of'
    elif [ $1 = '-r' ]
    then
        POSITION='right-of'
    elif [ $1 = '-u' ]
    then
        POSITION='above'
    else
        echo "-l Display piccolo a sinisitra"
        echo "-r Display piccolo a destra"
        echo "-l Display piccolo sopra"
        exit 1
    fi
fi
```



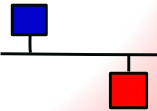
Clonazione e scelta schermi

- Per clonare gli schermi ho utilizzato la funzione **setClone** già presentata nelle precedenti slide
- Per individuare lo schermo primario ho utilizzato lo stesso metodo delle precedenti slide
- Il problema è distinguere il proiettore (**SECONDARY**) dallo schermo aggiuntivo (**TERTIARY**)
- Nella riga con il nome dello schermo c'è anche , dopo il **connected** e l'eventuale **primary**, la risoluzione attuale
- Userò come **SECONDARY** quello con risoluzione maggiore e come **TERTIARY** quello con risoluzione minore.



Individuazione dei tre monitor

- Per individuare il primario uso il comando delle precedenti slide
PRIMARY=`xrandr | grep " connected" | grep primary | sed -e "s/ .\*\$/\"`
- Per individuare il secondario ed il terziario creo per prima cosa un *array* delle risoluzioni (la sola larghezza) indicate nella riga del monitor
- Per fare questo utilizzerò la funzione **declare** (che è un sinonimo del comando **typeset** usato nella funzione)
- Uso questa sequenza di comandi:
declare R=(\$(xrandr | grep " connected" | grep -v primary | sed -e "s/\([^\]*\) c\([^\]*\) \([0-9]*\) .*\$/\2/"))
- Dichiariamo la variabile **R** che è un vettore (le parentesi tonde più esterne) riempito con il comando (la coppia **\$(e)**)
- Il comando è simile ai soliti: risultato di **xrandr**, selezione dei **connected** che non siano **primary**



La sostituzione in sed

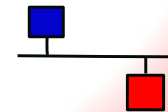
- Cambia la sostituzione nel comando **sed**:

```
sed -e "s/[ ^]* c[ ^]* \([0-9]*\).*$/\1/"
```

- Vediamo per primo un esempio della riga da elaborare:

HDMI-1-0 connected 1920x1080+1920+0 (...)

- **"\([^]*\)"** cerca una sequenza di caratteri diversi da spazio – trova **"eDP-1 "**
- **"c[^]*"** cerca **c** seguito da una sequenza di caratteri diversi da spazio e quindi seguita da uno spazio – trova **"connected "** - non ci sarà un **primary** perché la precedente **grep** lo ha eliminato
- **\([0-9]*\)** cerca una sequenza di cifre (nel nostro caso la larghezza dello schermo) le tonde definiscono un **gruppo** e le \ prima delle tonde servono per non interpretarle come gruppo di comandi – trova **"1920"**
- **".*"** raccoglie tutti i caratteri fino alla fine della stringa (fa pulizia)
- **\1** Nella sostituzione fa riferimento al primo **gruppo** e quindi restituisce **"1920"**

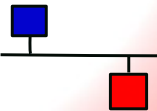


Cerchiamo la risoluzione minore

- Per trovarla basta un semplice if che scelga tra i due elementi del vettore **R**
- La risoluzione minore sarà inserita in **SRES**:

```
if expr ${R[0]} \< ${R[1]}
then
    SRES=${R[0]}
else
    SRES=${R[1]}
fi
```

- Si noti la \ prima del <, per evitare che < sia interpretato come redirectione
- Inoltre utilizziamo il comando **expr** che fa confronti numerici e non di testo come [



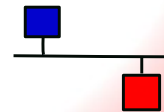
Individuiamo i due monitor

- Per trovare il **SECONDARY** ed il **TERTIARY** useremo la risoluzione minore come discriminante, quindi il **SECONDARY** sarà quello che **NON** ha la risoluzione **SRES**, mentre il **tertiary** sarà quello che ce l'ha:

```
SECONDARY=`xrandr | grep " connected" | grep -v primary |  
grep -v ${SRES}x | sed -e "s/ .*$//"`  
TERTIARY=`xrandr | grep " connected" | grep -v primary | grep ${SRES}x |  
sed -e "s/ .*$//"`
```

- Purtroppo questa non funziona per **SECONDARY** se ha la stessa risoluzione del **TERTIARY**.
- Per ovviare, se **SECONDARY** è vuoto, cercherò il monitor non primario con risoluzione **SRES** e nome diverso da **TERTIARY**:

```
if [ -z $SECONDARY ]  
then  
    SECONDARY=`xrandr | grep " connected" | grep -v primary | grep  
                ${SRES}x | grep -v $TERTIARY | sed -e "s/ .*$//"`  
fi
```



Inviemo i comandi

- Per finire, potremo inviare i comandi, utilizzando la funzione per la clonazione e xrandr direttamente per il monitor terziario:

```
xrandr | setClone $PRIMARY $SECONDARY  
xrandr --output $TERTIARY --$POSITION $PRIMARY
```

