

## Aplicacions d'interacció amb l'usuari: Zenity i Dialog

### **Zenity:**

Aquest programa normalment ve instal·lat de sèrie amb l'entorn Gnome (el seu equivalent a KDE seria el Kdialog i l'equivalent en Ncurses -que veurem després- seria el Dialog), i és el programa responsable de mostrar diferents tipus de quadres de diàlegs interactius. Pots trobar molts exemples d'ús a <http://library.gnome.org/users/zenity/stable>

### Generalitats

Zenity sempre retorna un dels següents valors, tots ells recollits per la variable especial \$? Normalment això no ens caldrà saber-ho excepte quan utilitzem el quadre --question.

- 0 -> L'usuari ha pulsat el botó d'OK ó de Tancar
- 1 -> L'usuari ha pulsat el botó de Cancel·lar (o el botó de tancar de la barra de control)
- 1 -> Ha ocorregut un error inesperat
- 5 -> El quadre s'ha tancat perquè el temps d'espera ha sigut sobrepassat

Tots els tipus de quadres Zenity tenen les opcions generals següents:

--title = "El títol del quadre"  
--width = *amplada\_en\_px\_del\_quadre*  
--height = *alçada\_en\_px\_del\_quadre*  
--timeout = *segons\_despera\_despres\_dels\_quals\_el\_quadre\_es\_tanca*  
--window-icon = *"ruta/de/la/icona/mostrada/al/quadre"* També es pot especificar, en comptes d'una ruta, les paraules "info", "warning", "question" ó "error", de manera que aparegui una icona ja predefinida corresponent a cadascun d'aquests quatre casos.

### Quadres bàsics d'entrada i sortida

Zenity pot servir per mostrar un missatge d'error:  
*zenity --error --text="No es pot trobar /var/log/syslog."*

O una advertència:  
*zenity --warning --text="Desconnecteu el cable de corrent per evitar problemes"*

O bé informatiu:  
*zenity --info --text="S'han actualitzat 3 de 10 fitxers"*

El text pot incloure els caràcters d'escapament típics, com \n pel salt de línia ó \t pel tabulador

Zenity també pot servir per mostrar una finestra emergent de notificació (la icona indicada pot ser "error", "warning", "info" o "question"):  
*zenity --notification --window-icon="info" --text="Actualització necessària!"*

En aquest cas també podríem fer servir la comanda *notify-send*, així:  
*notify-send "Títol" "Missatge"*

Aquesta comanda admet el paràmetre -t n.º per indicar el número de segons que romandrà

visible la finestreta (si es posa 0 serà permanent). També admet el paràmetre `-u {low|normal|critical}` que serveix per canviar la icona mostrada (similars als d'error, warning, info, respectivament). Finalment, el paràmetre `-i /ruta/absoluta/imatge` serveix per mostrar la imatge (png, ico, jpg) indicada juntament amb el missatge.

El missatge pot venir acompanyat de diferents etiquetes de format a l'estil HTML com ara `<u>...</u>` (subratllat), `<b>...</b>` (negrita), `<i>...</i>` (cursiva) o `<s>...</s>` (ratllat)

Zenity pot servir per mostrar una quadre de pregunta Sí/No:

```
zenity --question --text="Estàs segur de continuar?"
```

En aquest cas per saber quina ha sigut la resposta de l'usuari hem de conèixer el valor que ha assolit la variable \$?, així, per exemple:

```
zenity --question --text "Segur que vols apagar l'ordinador?"; echo $?
```

Zenity pot servir per mostrar un quadre d'entrada de text, fent servir l'opció `--entry`.

Opcionalment es poden afegir a `--entry` aquestes opcions específiques:

`--text = "Missatge explicatiu que es mostra"`

`--entry-text = "Text mostrat per defecte al camp d'entrada"`

`--hide-text` (oculta el text al camp d'entrada del quadre)

Zenity retorna el contingut de l'entrada de text, així que podríem utilitzar una variable per tal d'emmagatzemar la resposta de l'usuari (si ha pulsat "Acceptar", és clar...si pulsa "Cancel·lar" no s'emmagatzema res) per utilitzar-la posteriorment. Així, per exemple:

```
resposta=$(zenity --entry --text "On estàs?"); echo $resposta
```

Zenity pot servir per mostrar un quadre d'informació textual (en principi de només lectura) fent servir l'opció `--text-info`. El text a mostrar pot provenir o bé de la sortida d'una altra comanda entubada (com per exemple, així)...:

```
ps ax | zenity --text-info --width 530
```

...o bé provenir d'un fitxer de text, el qual s'indicarà amb el paràmetre `--filename`, així:

```
zenity --text-info --filename = "nom_del_fitxer_que_conté_el_texte_mostrat"
```

Si es vol que el contingut mostrat sigui editable, cal afegir el paràmetre `--editable`. Zenity retorna el text editat (quan el quadre es tanca), així que podríem utilitzar una variable per tal d'emmagatzemar la resposta de l'usuari per utilitzar-la posteriorment, de forma similar a amb `--entry`.

Zenity pot servir per mostrar un quadre de llista per seleccionar ítems (a mode de "menú d'opcions"), fent servir l'opció `--list`; en aquest cas es retorna el valor de la primera columna corresponent a les files seleccionades, així que podríem utilitzar una variable per tal d'emmagatzemar la resposta de l'usuari per utilitzar-la posteriorment. Les opcions més importants són:

`--column = "Títol de columna"` (s'ha d'escriure una opció `--column` per cada columna que es vol mostrar. Per indicar els valors corresponents a les diferents columnes d'una fila, s'han d'escriure un rera l'altre, separats per espais en blanc i així amb totes les files que calgui)

`--checkboxlist` (especifica que la primera columna conté check boxes)

`--radiolist` (especifica que la primera columna conté botons radio)  
(Si s'utilitza `--checklist` ó `--radiolist`, el valor de la 1ª columna haurà de ser 'TRUE' ó 'FALSE')  
`--editable` (permet l'edició dels ítems de la llista)  
`--separator` = "cadena\_que\_fara\_de\_separador\_si\_es\_retorna\_més\_dun\_item"  
`--print-column` = nº\_columna\_que\_conté\_el\_valor\_retornat. (per defecte és 1; es pot utilitzar la constant ALL per retornar el valor de totes les columnes)

Un exemple de llista estàndar:

```
res=$(zenity --list --column "Nom" --column "Cognom" Luis Pérez Ana Gómez)
echo $res
```

Un exemple de llista radio:

```
res=$(zenity --list --text "Messi és guapo?" --radiolist --column "Escull" --column "Opinió"
TRUE Moltíssim FALSE Bastant FALSE Normalet FALSE "És lletgíssim")
echo $res
```

Un exemple de llista checkbox:

```
res=$(zenity --list --text "Digues els teus plats preferits" --checkboxlist --column "Escull"
--column "Plats" TRUE "Macarrons" TRUE "Cols amb espinacs" FALSE "Patates fregides"
FALSE "Butifarra amb mongetes" --separator=":")
echo $res
```

### Altres quadres

Zenity pot servir per mostrar un calendari, fent servir l'opció `--calendar`. A més, es tenen aquestes opcions específiques:

`--text` = "Text que es mostra"  
`--day` = nº\_dia\_seleccionat\_per\_defecte (entre 1 i 31; si no s'indica, s'agafa el dia actual)  
`--month` = nº\_mes\_seleccionat\_per\_defecte (entre 1 i 12; si no s'indica, s'agafa el mes actual)  
`--year` = nº\_any\_seleccionat\_per\_defecte (si no s'especifica, s'agafa l'any actual)  
`--date-format` = format\_de\_data\_retornat\_un\_cop\_shagi\_seleccionat.  
(El format per defecte depèn de la configuració regional local. El format que s'especifiqui ha de ser acceptable, com per exemple: %A %d/%m/%y)

Zenity retorna la data seleccionada, així que podríem utilitzar una variable per tal d'emmagatzemar la resposta de l'usuari per utilitzar-la posteriorment. Així, per exemple:

```
data=$(zenity --calendar --text "Selecciona dia" --title "Cita" --day 23 --month 5 --year 2014) ;
echo $data
```

Zenity pot servir per mostrar un quadre de selecció de fitxers, fent servir l'opció `--file-selection`. A més, es tenen aquestes opcions específiques:

`--filename` = nom\_arxiuodirectori\_seleccionat\_per\_defecte  
`--multiple` (permet la selecció de múltiples arxius)  
`--directory` (només permet la selecció de directoris)  
`--save` (posa el mode del quadre en "save")  
`--separator` = "cadena\_que\_fa\_de\_separador\_a\_la\_llista\_retornada\_de\_noms\_de\_fitxers"

Zenity retorna les rutes dels fitxer ó directoris seleccionats, així que podríem utilitzar una variable per tal d'emmagatzemar la resposta de l'usuari per utilitzar-la posteriorment. Així:

```
rutes=$(zenity --file-selection --save --confirm-overwrite); echo $rutes
```

Zenity pot servir per mostrar una barra horitzontal (un "slider") per seleccionar fàcilment un valor numèric dins d'un rang, fent servir l'opció `--scale`, així:

```
res=$(zenity --scale --text "Escull" --min-value=2 --max-value=100 --value=50 --step 2)  
echo $res
```

Alguns quadres Zenity suporten l'ús de “tecles d'accés”. Per especificar el caràcter a utilitzar com a tecla d'accés, s'ha d'escriure un guió baix abans del caràcter en qüestió que aparegui al text del quadre.

Un programa alternatiu a Zenity (però basat en ell) que ofereix més flexibilitat i versatilitat és Yad (<https://yad-dialog.sourceforge.io>).

## Dialog:

La diferència fonamental entre el Dialog (<http://invisible-island.net/dialog>) i el Zenity és que el primer no necessita cap entorn gràfic per funcionar (perquè està basat en la llibreria Ncurses en comptes de la llibreria GTK+). És possible, no obstant, que aquest programa no estigui instal·lat per defecte; en tot cas, sempre és bo comprovar-ho primer abans de fer-lo servir en un script (per exemple controlant la sortida de la comanda *which dialog*) i, si no ho estigués, instal·lant el paquet corresponent (anomenat "dialog").

A continuació es presenten alguns exemple d'ús, força autoexplicatius (els dos darrers números que apareixen a moltes comandes són l'alçada i l'amplada i del quadre on s'imprimirà el missatge indicat. Si l'amplada fos menor que la longitud de la cadena a mostrar, el text d'aquesta s'anirà adaptant però si el tamany del quadre és massa petit, els caràcters sobrants no es veuran).

**NOTA:** La tecla TAB ens servirà per moure'ns per les diferents opcions/botons del quadre mostrat, la tecla ENTER ens servirà per seleccionar-la efectivament i la tecla SPACE ens servirà, si hi haguessin opcions a escollir (les quals es poden recórrer amb els cursors), per marcar-les/desmarcar-les.

```
dialog --title "Títol" --msgbox "Hola amic" 5 20
```

També es pot fer servir el paràmetre *--backtitle* en comptes de *--title* (l'efecte és diferent)

```
dialog --title "Títol" --infobox "Espera si's plau" 10 30 ; sleep 4
```

Similar al *--msgbox* però sense botó de tancar el quadre: aquesta acció és temporitzada

```
dialog --title "Títol" --yesno "Ets guapo?" 6 25
```

El valor de \$? serà 0 si la resposta es "Sí" i 1 si la resposta és "No"

```
dialog --title "Títol" --textbox /ruta/arxiu/texte/a/mostrar 22 70
```

```
dialog --title "Títol" --inputbox "Escriu el teu nom:" 8 40 2>resposta.txt
```

Fixar-se que el que escrigui l'usuari es redirecciona a la sortida d'error, no la estàndar.

**NOTA:** Per a què es redireccioni a la sortida estàndar (i així, per exemple, es pugui guardar aquest valor en una variable amb la sintaxi *\$(dialog ...)*, s'ha d'afegir el paràmetre *--stdout*

```
dialog --title "Títol" --menu "Hola" 10 30 20 id1 Cadena1 id2 Cadena2 [...] 2> resp.txt
```

El tercer número és l'alçada que tindrà el menú dins del quadre. Els valors "idX" poden ser qualsevol cosa però hauran de ser únics perquè és el que s'envia per la sortida d'error (per tal d'identificar quina opció ha sigut l'escollida). Igualment, té el paràmetre *--stdout*

```
dialog --title "Títol" --checkboxlist "Hola" 10 30 20 id1 Cadena1 {on|off} [...] 2> resp.txt
```

El valor "on/off" és l'estat inicial del checkbox. Igualment, té el paràmetre *--stdout*

```
dialog --title "Títol" --radiolist "Hola" 10 30 20 id1 Cadena1 {on|off} [...] 2> resp.txt
```

Igual que el quadre "checkboxlist" però ara només es pot escollir un únic valor, no varis a la vegada. Igualment, té el paràmetre *--stdout*

```
dialog --title "Títol" --fselect "/ruta/carpeta/" 40 70
```

Mostra el quadre de selecció de fitxers (obrint-se a la carpeta indicada). Retorna la ruta del fitxer seleccionat (per defecte a la sortida d'error...per a què ho faci a l'estàndar s'ha d'afegir el paràmetre *--stdout*

Es poden trobar més exemples a <http://invisible-island.net/dialog/dialog-figures.html>

A continuació es presenta un exemple d'script pràctic on s'utilitza *dialog*: es tracta d'un programa que realitza una còpia de seguretat de l'escriptori de l'usuari actual (guardant-la comprimida en format tgz dins de la seva pròpia carpeta personal); si executéssim aquest script com una tasca programada del sistema (aprendrem a fer-ho més endavant) tindríem una manera automàtica de realitzar còpies sense haver d'utilitzar cap programa extern.

```
#!/bin/bash
dialog --title "Backup" --msgbox "\nBenvingut a l'aplicatiu de còpia de seguretat de l'Escriptori.\n\nPulsa
<Enter> per començar o <Esc> per sortir" 10 70
if [[ "$?" != "0" ]] # Un valor de $? diferent de 0 indica 'Cancel'
then
    dialog --title "Backup" --msgbox "\nCòpia cancelada" 10 30
else
    dialog --title "Backup" --infobox "\nCòpia en procés..." 10 30
    #La comanda tar -czf genera un paquet tgz (el nom/ruta del qual s'indica com a primer paràmetre) amb el
    contingut de la carpeta indicada com a 2n paràmetre. Per restaurar s'hauria d'anar a la carpeta on volguem
    restaurar-lo (amb cd) i allà executar la comanda: tar -xzf /ruta/paquet.tgz
    cd ~ ; tar -czf "./copiaEscriptori-$(date +%F_%T).tgz" "./Escriptori" 2> /tmp/errorcopia
    # Un valor de $? igual a 0 indica que la còpia ha sigut realitzada amb èxit
    if [[ "$?" == "0" ]]
    then
        dialog --title "Backup" --msgbox "\nCòpia completada amb èxit" 10 30
    else
        dialog --title "Backup" --msgbox "\nCòpia fallida. Pulsa <Enter> per veure l'error" 10 70
        dialog --title "Error Log" --textbox /tmp/errorcopia 20 80
    fi
fi
rm -f /tmp/errorcopia ; clear
```

A l'script anterior hem suposat que l'escriptori es trobava a la carpeta "~/Escriptori" però, ¿com podem tenir la garantia de què sigui així? El nostre script es podria executar en un ordinador on aquesta carpeta fos "~/Escritorio" o "~/Desktop", etc. La manera de saber quin és el nom d'aquesta carpeta és consultar el contingut de l'arxiu ~/.config/user-dirs.dirs ,el qual sol ser semblant a aquest:

```
XDG_DESKTOP_DIR="$HOME/Desktop"
XDG_DOWNLOAD_DIR="$HOME/Downloads"
XDG_TEMPLATES_DIR="$HOME/Templates"
XDG_PUBLICSHARE_DIR="$HOME/Public"
XDG_DOCUMENTS_DIR="$HOME/Documents"
XDG_MUSIC_DIR="$HOME/Music"
XDG_PICTURES_DIR="$HOME/"
XDG_VIDEOS_DIR="$HOME/Videos"
```

Per tant, executant una comanda com: *grep "XDG\_DESKTOP\_DIR" ~/.config/user-dirs.dirs | sed "s/XDG\_DESKTOP\_DIR=/"* obtindríem la ruta de l'escriptori actual sense problemes de traducció.

**NOTA:** Un arxiu que també és interessant és ~/.config/user-dirs.locale, el qual indica la locale en la què estan escrites les rutes dins de ~/.config/user-dirs.dirs . En qualsevol cas, si es fa algun canvi en algun d'aquests dos fitxers (en el primer per forçar la traducció del segon o en el segon directament per simplement canviar la ruta indicada per una altra), caldrà executar la comanda *xdg-user-dirs-update --force* per a què es facin efectius. Per més informació podeu consultar [https://wiki.archlinux.org/index.php/XDG\\_user\\_directories](https://wiki.archlinux.org/index.php/XDG_user_directories)