

## ncurses

### Utilisation des fenêtres avec ncurses

Cet article présente comment créer une fenêtre avec ncurses et créer une sous-fenêtre.

## 1 Créer une fenêtre simple

Dans cette première application, nous créons une fenêtre colorée. L'utilisateur doit cliquer dans celle-ci pour que le programme s'arrête.

Pour définir notre fenêtre, nous utilisons 4 constantes définies comme suit :

```
#define LARGEUR 20      /* Largeur de la fenêtre */
#define HAUTEUR 10      /* Hauteur de la fenêtre */
#define POSX 20         /* Position horizontale de la fenêtre */
#define POSY 5          /* Position verticale de la fenêtre */
```

Après avoir initialisé ncurses, nous procédons aux vérifications d'usage sur les dimensions du terminal.

```
if((COLS < POSX + LARGEUR) || (LINES < POSY + HAUTEUR)) {
    ncurses_stopper();
    fprintf(stderr,
        "Les dimensions du terminal sont insuffisantes : l=%d, h=%d au lieu de l=%d, h=%d\n",
        COLS, LINES, POSX + LARGEUR, POSY + HAUTEUR);
    exit(EXIT_FAILURE);
}
```



La création d'une fenêtre en dehors des limites ne génère pas d'erreur.

La création de la fenêtre est réalisée avec la fonction `newwin`. Nous ajoutons également un fond coloré pour l'identifier, sans oublier d'appeler `wrefresh` pour rafraîchir l'affichage.

```
fenetre = newwin(HAUTEUR, LARGEUR, POSY, POSX);
wbkgd(fenetre, COLOR_PAIR(2));
wrefresh(fenetre);
```

Pour finir, nous attendons que l'utilisateur clique dans la fenêtre pour que le programme s'arrête.

```
int quitter = FALSE;
while(quitter == FALSE) {
    i = getch();
    if((i == KEY_MOUSE) &&
        (souris_getpos(&sourisX, &sourisY, NULL) == OK))
        if((sourisX >= POSX) && (sourisX < POSX + LARGEUR) &&
            (sourisY >= POSY) && (sourisY < POSY + HAUTEUR))
            quitter = TRUE;
}
```

Une fois que l'utilisateur clique dans la fenêtre, le programme s'arrête. Il ne faut pas oublier de supprimer la fenêtre, puis d'arrêter `ncurses`.

```
delwin(fenetre);
ncurses_stopper();
```

## 2 Gérer le cadre d'une fenêtre

Dans le programme précédent, si la fenêtre n'est pas colorée, l'utilisateur ne peut pas la voir. `ncurses` propose de créer un cadre automatiquement. Par contre, si on efface la fenêtre ou si l'affichage provoque le défilement de la fenêtre, le cadre est modifié. Nous proposons ici de créer une fenêtre principale avec un cadre et une sous-fenêtre dans laquelle le texte sera affiché.

La création de la fenêtre principale et de la sous-fenêtre est réalisé avec le code suivant. Le cadre est créé à l'aide de la fonction `box`.

```
fenetre = newwin(HAUTEUR, LARGEUR, POSY, POSX);
box(fenetre, 0, 0);
sous_fenetre = subwin(fenetre, HAUTEUR - 2,
                      LARGEUR - 2, POSY + 1, POSX + 1);
scrollok(sous_fenetre, TRUE);
```

Dans la boucle principale, nous attendons que l'utilisateur clique dans la sous-fenêtre pour afficher un message dans celle-ci. Au bout d'un moment, le texte va défiler, mais le cadre restera inchangé.

```
while((i = getch()) != KEY_F(2)) {
    if((i == KEY_MOUSE) &&
        (souris_getpos(&sourisX, &sourisY, NULL) == OK))
        if((sourisX > POSX) && (sourisX < POSX + LARGEUR - 1) &&
            (sourisY > POSY) && (sourisY < POSY + HAUTEUR - 1)) {
            cpt++;
            if(cpt != 1) wprintw(sous_fenetre, "\n");
            wprintw(sous_fenetre, "Ca_chatouille_%d", cpt);
            wrefresh(sous_fenetre);
        }
}
```

Une fois que l'utilisateur presse la touche F2, le programme s'arrête. Il ne faut pas oublier de supprimer la sous-fenêtre et la fenêtre puis d'arrêter `ncurses`.

```
delwin(fenetre);  
delwin(sous_fenetre);  
ncurses_stopper();
```

### 3 Compilation et exécution

Le `makefile` fourni permet de compiler les programmes précédent. Tapez la commande suivante :

```
make
```

Pour exécuter les programmes, tapez l'une des commandes suivantes :

```
./exemple1  
./exemple2
```