



RECURSOS

programação sistemática

# O que é?

Biblioteca para C que lida com interface textual de terminais.

The screenshot shows the YaST2 terminal interface with the following components:

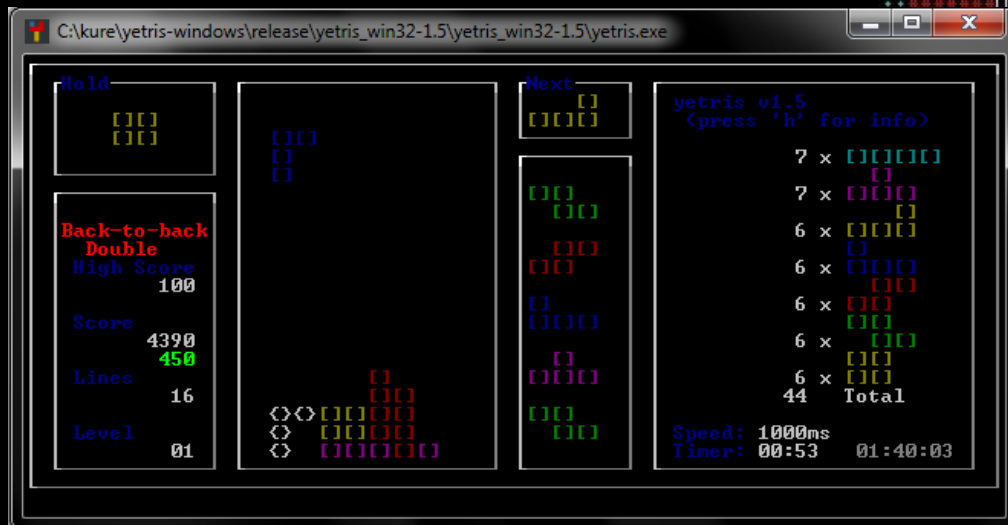
- Header:** YaST2 - sw\_single @ bourbaki
- Navigation:** [Dependencies] [View] [Configuration] [Extras]
- Filter Section:**
  - Search:
  - Search Phrase:
  - [x] Ignore Case
  - Search Mode:
- Package List:**

	Name	Summary
	<b>autoyast2</b>	<b>YaST2 - Auto</b>
i	autoyast2-installation	YaST2 - Auto
	openwsman-yast	YaST plugin
	patterns-openSUSE-devel_yast	Meta package
	patterns-openSUSE-gnome_yast	Meta package
	patterns-openSUSE-kde4_yast	Meta package
	patterns-openSUSE-x11_yast	Meta package
	patterns-openSUSE-yast2_basis	Meta package
	patterns-openSUSE-yast2_install_wf	Meta package
i	yast2	YaST2 - Main
- Summary:** 203 packages found
- Package Details:**
  - Package: autoyast2
  - autoyast2 - YaST2 - Automated Installation
  - Version: 2.18.14-1.1.3
- Actions:** [Actions] [Cancel] [Accept]
- Bottom Panel:** config - Linux Kernel v2.6.32 Configuration
  - General setup --->
  - [\*] Inable loadable module support --->
  - Inable the block layer --->
  - Processor type and features --->
  - Power management and ACPI options --->
  - Bus options (PCI etc.) --->
  - Executable file formats / Emulations --->
  - Networking support --->
  - Device Drivers --->
  - Firmware Drivers --->
  - File systems --->



# Por que?

- Interface melhor que linha de comando
- Bastante utilizada
- *Cross-platform*
- Vai cair no trabalho





# Instalação

Linux:

```
$ sudo apt-get install libncurses5-dev
```

Windows:

PD curses



# Primeiro programa!

```
#include <ncurses.h>
```

```
int main()
```

```
{
```

```
    initscr();        /* inicia ncurses    */
```

```
    move(3, 2);        /* move o cursor    */
```

```
    addch('x');        /* adiciona um char */
```

```
    getch();           /* espera um input  */
```

```
    endwin();          /* finaliza ncurses */
```

```
    return 0;
```

```
}
```

```
$ gcc primeiro.c -o primeiro -lncurses
```



# Primeiro programa (melhorado)!

```
#include <ncurses.h>
int main()
{
    initscr();
    start_color();                /* ativa cores */
    init_pair(1, COLOR_RED, COLOR_BLACK); /* inicia par "1" */
    attrset(COLOR_PAIR(1));      /* ativa par "1" */
    mvaddstr(3, 2, "Hello, World!"); /* mostra string */
    getch();
    endwin();
    return 0;
}
```

```
$ gcc segundo.c -o segundo -lncurses
```



# Terceiro programa

```
#include <ncurses.h>

int main()
{
    int sair=0, x=3, y=3;
    initscr();
    cbreak();
    noecho();
    timeout(0);
    while (sair == 0)
    {
        switch(getch())
        {
            case 'a': x--; break;
            case 'd': x++; break;
            case 'w': y--; break;
            case 's': y++; break;
            case 'q': sair = 1; break;
        }
        mvaddch(y, x, 'x');
    }
    endwin();
    return 0;
}
```

/\* inicia ncurses \*/

/\* recebe input sem precisar apertar <enter> \*/

/\* não mostra na tela input recebido \*/

/\* não fica esperando por input \*/

/\* "wasd" controlam o cursor \*/

/\* mostra char na posicao x, y \*/

```
$ gcc terceiro.c -o terceiro -lncurses
```