



UNIVERSITÀ  
DI PAVIA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL' INFORMAZIONE

# Da dove nasce il terminale di Linux

Tullio Facchinetti

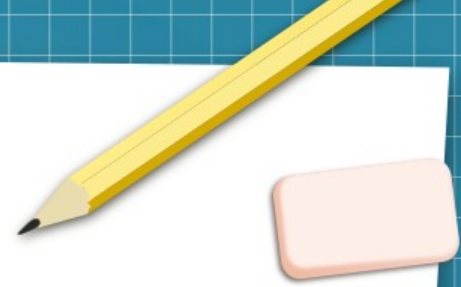
# Sistemi operativo



Il **sistema operativo** è il programma/software che gestisce le risorse hardware e software della macchina, fornendo servizi di base ai software applicativi.

Abbreviato in SO, in inglese OS (Operating System).

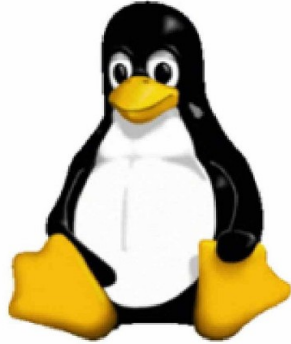
# Sistemi operativo



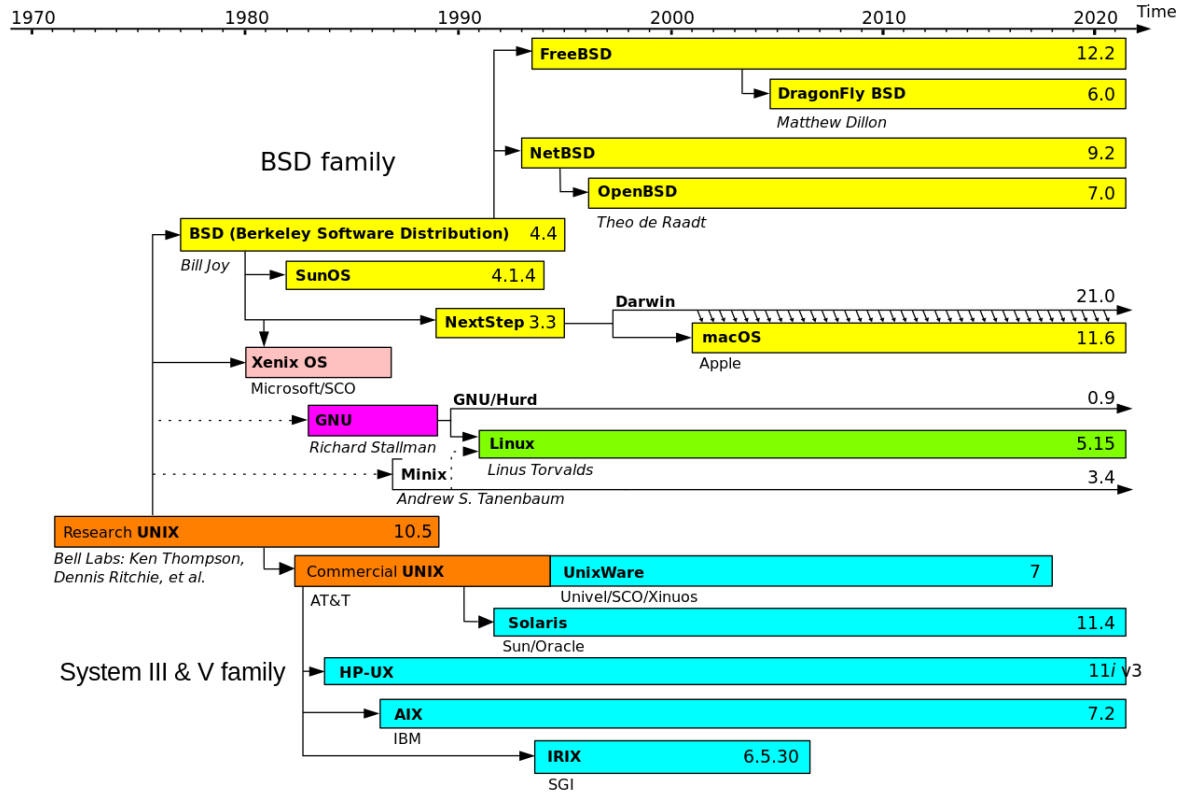
Composto da più sottosistemi o componenti software:

- Kernel - Funzionalità di base e comuni
- Scheduler - Gestione dei processi (programmi)
- Gestore della memoria
- Gestore delle periferiche - scheda grafica, scheda di rete (es. WiFi)
- Spooler di stampa - Coda di stampa
- **Interfaccia utente** - Finestre, emulatore di terminale
- **File system** - Gestione di file e directory

# I sistemi operativi

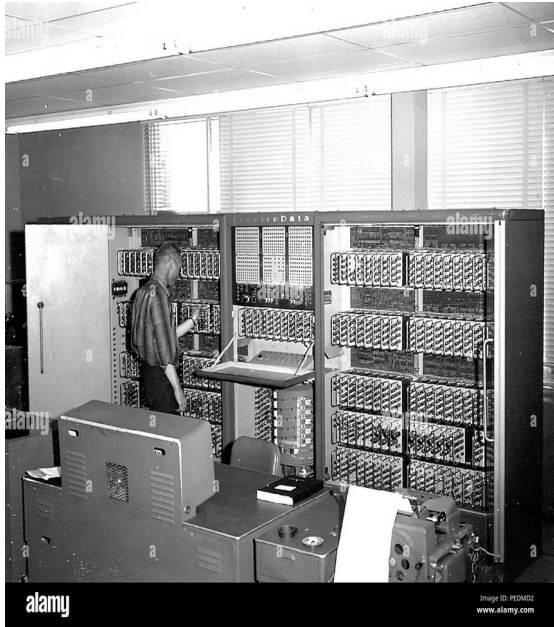


# Origine di UNIX / Linux



# Nascita del terminale

## Mainframe



## Terminali



```
Some Utilities  Computers  Options  System  Help
-----
ISP Primary Option Menu

Option ---->  Name
0 Settings Terminal and user parameters User ID : 207101
1 View Display source data or settings Line : 07:52
2 Edit Create or change source data Terminal : 0278
3 Utilities Perform utility function Screen : 128
4 Foreground Interactive language processing Language : ENGLISH
5 Batch Select job for language processing Job ID : 128
6 Command Enter ISD or Workstation commands ISD Input : DDPH002
7 Startup Test Perform startup testing ISD output : 07:111
8 Log Facility Library administrator functions Job ID : 5001
9 IBM Product IBM program development products Job acct : 750
10 SCL SCL Command Question Library Manager ISD Object/Action Interface Release : ISP 7.3
11 Workplace

----- Other Functions -----
00 SDDP View output
01 Data Edit Shell (prompt)
Enter X to terminate using log/list defaults
04:01
```



```
"TERMINAL GREEN IS FOR THOSE OF
US WHO WANT TO DRESS LIKE A SYSTEM
BEHIND A SYSTEM." █
```



Interfaccia  
a caratteri

## Stampante

# Il terminale



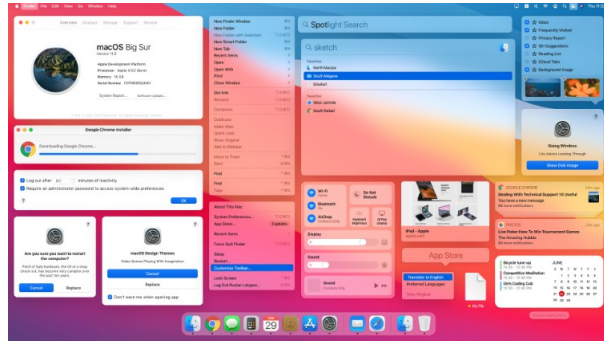
Oggigiorno i terminali hardware sono poco usati; mentre sono molto comuni gli **emulatori di terminale**.

**Componente hardware** usato per inserire dati in **ingresso** ad un computer e riceverli in **uscita** (output) per la loro visualizzazione.



# Interfacce utente

**GUI** - Graphical User Interface  
Finestre, immagini, ecc.



**CLI** - Command Line Interface  
Solo testo e comandi

```
ls -l /software
```

Name	Size	MTime	Name	Size	MTime
..	4096	Oct 2 04:02	..	4096	Oct 2 04:02
/ICAClient-3.0	2048	Jan 6 2003	/.java	30	May 13 2004
/aida-2.1.1	2048	Apr 28 2003	/ada	4096	Aug 9 2001
/amber-6.0	2048	Feb 27 2004	/conf	151	Jul 19 2000
/amber-7.0	2048	Mar 5 2004	/config	4096	Dec 13 2004
/amber-7.0p	2048	Apr 16 2004	/cron.d	133	Sep 29 20:23
/amber-9	2048	Dec 22 2004	/default	75	Aug 12 2004
-ansys61	34	Jan 7 2003	/dt	27	Apr 5 2003
-ansys71	34	Nov 28 2003	/Zskloggs	39	Aug 3 2000
/ant-1.6	2048	Aug 10 13:26	-fstyp.d	15	Apr 25 2000
/apache-1.3.27	2048	Dec 16 2002	-httpd	20	Jul 19 2000
/apache-1.3.28	2048	Jan 6 2004	/init.d	4096	Sep 21 15:45
/apache-1.3.33	2048	Feb 7 2005	/js	4096	Aug 9 2001
/autoconf-2.57	2048	May 27 2004	/lost+found	4096	Oct 8 2004
/autodock-305	2048	Jan 5 2001	/mail	4096	May 2 10:04
/ICAClient-3.0			/cron.d		

Hint: Keys not working in xterms? Use our xterm.ad, .ti and .tcap files.

```
alf@software~$
```

**TUI** - Text User Interface  
Solo testo ma con elementi da GUI

```
alf@server ~]$ netstat -atu
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:ssh	0.0.0.0:*	LISTEN
tcp	0	0	server:ssh	192.168.1.48:59476	ESTABLISHED
tcp6	0	0	:::mysql	:::*	LISTEN
tcp6	0	0	:::ssh	:::*	LISTEN
udp	0	0	0.0.0.0:slingshot	0.0.0.0:*	
udp	0	0	0.0.0.0:7091	0.0.0.0:*	
udp	0	0	0.0.0.0:bootpc	0.0.0.0:*	
udp6	0	0	:::25087	:::*	
udp6	0	0	server:dhcpv6-client	:::*	
udp6	0	0	:::59809	:::*	

```
alf@server ~]$
```

# Emulatori di terminale

Ci sono molti emulatori di terminale.

Le funzionalità di base sono identiche: visualizzazione di testo e ricezione di input da tastiera.

Alcune differenze riguardano:

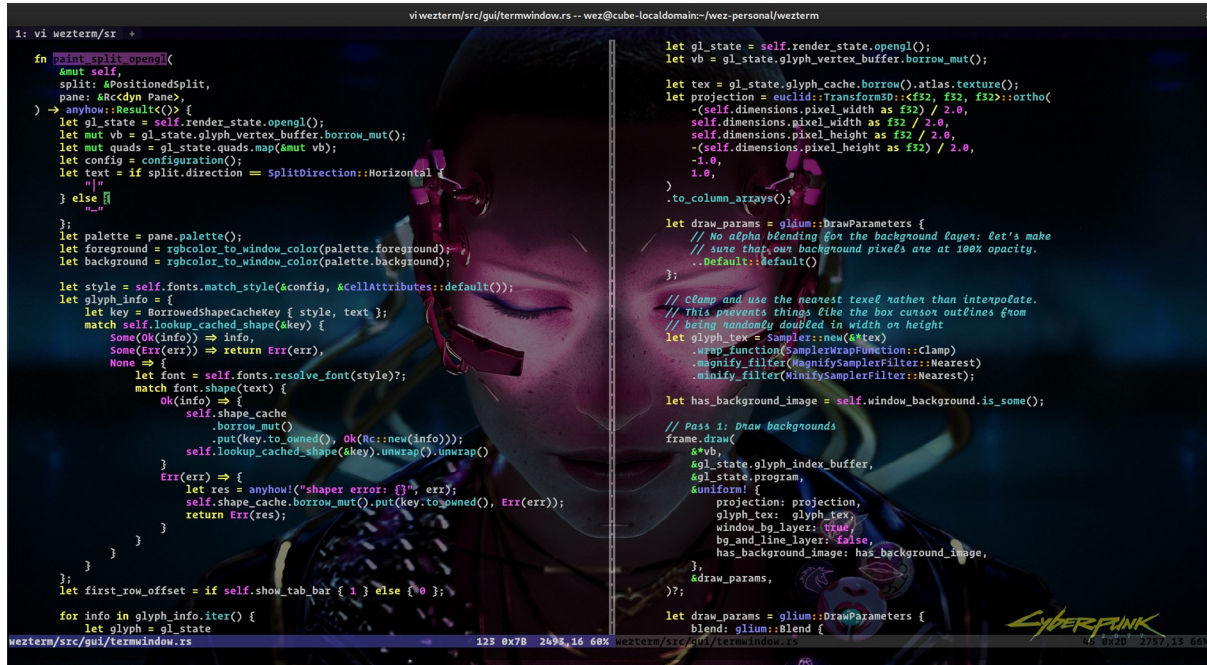
- Gestione di più finestre/tab
- Accelerazione grafica
- Gestione del mouse
- Inclusione di strumenti utili
- Portabilità

```
l@mac:~$ ls
videlo
l@mac:~$ cd ..
l@mac:~$ cd etc
l@mac:~$ ls
0,0,10,1maddr,arpa  csh_cshrc  gskadur-  logrotate.d  odbcinst.ini  rat
address.conf       csh_login  gskadur-  lynx.cfg    osonoffice   rps
adittime           dc_cshrc  host_conf  nagie       opt           screenc
aliases            debconf.conf  hostname  nallcp     pan.conf     security
alternatives       debconf     hosts      mailcap_order  passwd       servicos
apt                debian_version  hosts.allow  mailrc      passwd       shadu
asterisk           defoma     hotplug   nmapath_conf  perl         shadu-
at_deng           dcluser.conf  hotplug.d  noda       printcap     shells
bakplungfu        dclinet_conf  identd.conf  mediaprs   profile      skel
bash_bashrc       dclinet-script  identd_key  nline_tups  profile      ssh
bash_completion   dictionary     isns.conf   nmapath     python2.3    sudaba
bash_completion.d  discover_conf  init.d      nodprobe.d  rshdlib     sudoers.conf
bind              discover_conf-2,6  initdb     modules     rshlog.conf  sudoers
bikid,tab         discover.d     inputrc    modules.conf  rotd         syslog.conf
bikid,tab.old     dkg          ipkurgfu  modules.conf.old  rc1.d       termInfo
colmde            ewac21       issue      modules     rc2.d       timesync
chatscripts       email_addresses  issue_net  modutils   rc3.d       ucf_conf
chronolog.conf    environment   jkern:reg.conf  modutils   rc4.d       updateub.conf
complete,tcsh    exim         ldso.conf  ntools_conf  rc5.d       videlo,net_hosts
console          fdmount_conf  id,so,cache  nutils     rc6.d       vsh
cron              fonts        id,so.conf  nutils     rc7.d       wnetro
cron.d           fonttools    locale.alias  nutils     rc8.d       #wdal.conf#
cron.daily       ftpab        locale.gen   nutils     rc9.d       wdal.conf
cron.hourly      groff        locale.gen   network     reportbug.conf  wdal.conf#
cron.monthly     group        logcheck    network     resolv.conf   wdal.conf#
cron.quartz      grsync       login_defs   network     resolv.conf   wdal.conf#
cron.weekly      gshadu       logrotate.conf  odbc.ini    resolv.conf,pppd-backup  wslot
```

[https://en.wikipedia.org/wiki/List\\_of\\_terminal\\_emulators](https://en.wikipedia.org/wiki/List_of_terminal_emulators)

# Wezterm

Emulatore di terminale disponibile per diversi sistemi operativi



```
1: vi wezterm/src +
fn print_split_opengl(
    &mut self,
    split: &PositionedSplit,
    pane: &RC<dyn Pane>,
) -> anyhow::Result<> {
    let gl_state = self.render_state.opengl();
    let mut vb = gl_state.glyph_vertex_buffer.borrow_mut();
    let mut quads = gl_state.quads.map(&mut vb);
    let config = configuration();
    let text = if split.direction == SplitDirection::Horizontal
        {
            "1"
        } else {
            "-"
        };
    let palette = pane.palette();
    let foreground = rgbcolor_to_window_color(palette.foreground);
    let background = rgbcolor_to_window_color(palette.background);

    let style = self.fonts.match_style(&config, &cellAttributes::default());
    let glyph_info = {
        let key = borrowedShapeCacheKey { style, text };
        match self.lookup_cached_shape(&key) {
            Some(Ok(info)) => info,
            Some(Err(err)) => return Err(err),
            None => {
                let font = self.fonts.resolve_font(style)?;
                match font.shape(text) {
                    Ok(info) => {
                        self.shape_cache
                            .borrow_mut()
                            .put(key.to_owned(), Ok(Rc::new(info)));
                        self.lookup_cached_shape(&key).unwrap().unwrap()
                    }
                    Err(err) => {
                        let res = anyhow!("shaper error: {}", err);
                        self.shape_cache.borrow_mut().put(key.to_owned(), Err(err));
                        return Err(res);
                    }
                }
            }
        };
    };
    let first_row_offset = if self.show_tab_bar { 1 } else { 0 };
    for info in glyph_info.iter() {
        let glyph = gl_state

let gl_state = self.render_state.opengl();
let vb = gl_state.glyph_vertex_buffer.borrow_mut();
let tex = gl_state.glyph_cache.borrow().atlas.texture();
let projection = euclid::Transform3D::(<f32, f32, f32>::ortho(
    -(self.dimensions.pixel_width as f32) / 2.0,
    self.dimensions.pixel_width as f32 / 2.0,
    -(self.dimensions.pixel_height as f32) / 2.0,
    -(self.dimensions.pixel_height as f32) / 2.0,
    1.0,
    -1.0,
));
.to_column_arrays();

let draw_params = glium::DrawParameters {
    // No alpha blending for the background layer: let's make
    // sure that our background pixels are at 100% opacity.
    ..Default::default()
};

// Clamp and use the nearest texel rather than interpolate.
// This prevents things like the box cursor outlines from
// being randomly doubled in width on height
let glyph_tex = Sampler::new(&tex)
    .wrap_function(SamplerWrapFunction::Clamp)
    .magnify_filter(MagnifySamplerFilter::Nearest)
    .minify_filter(MinifySamplerFilter::Nearest);

let has_background_image = self.window_background.is_some();

// Pass 1: Draw backgrounds
frame.draw(
    &*vb,
    &gl_state.glyph_index_buffer,
    &gl_state.program,
    &uniform {
        projection: projection,
        glyph_tex: glyph_tex,
        window_bg_layer: true,
        bg_and_line_layer: false,
        has_background_image: has_background_image,
    },
    &draw_params,
);

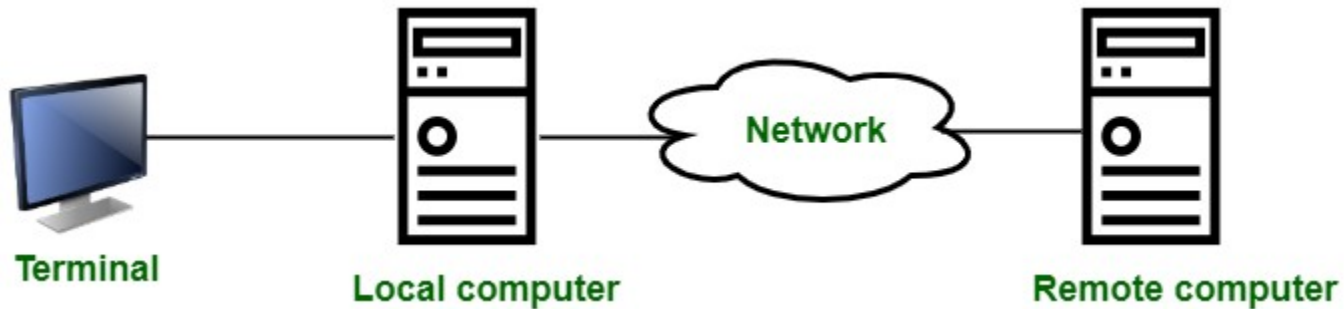
let draw_params = glium::DrawParameters {
    blend: glium::Blend {
```

# Collegamento remoto

Uno dei vantaggi del terminale è che permette di lavorare su un *computer remoto* **come se si fosse sul proprio** (*computer locale*).

Il computer locale si collega tramite la rete con il computer remoto.

Si usano tastiera e monitor (terminale) del computer locale, ma si utilizza il *file system* del computer remoto.



# Secure Shell (SSH)

- Protocollo di comunicazione per il collegamento tra computer
- Permette di collegare un terminale ad un computer remoto
- Comunicazione sicura -> crittografia e autenticazione
- Successore di Telnet (non sicuro)



# Collegamento remoto con SSH



Per un collegamento a computer remoto tramite SSH sono necessarie le seguenti informazioni:

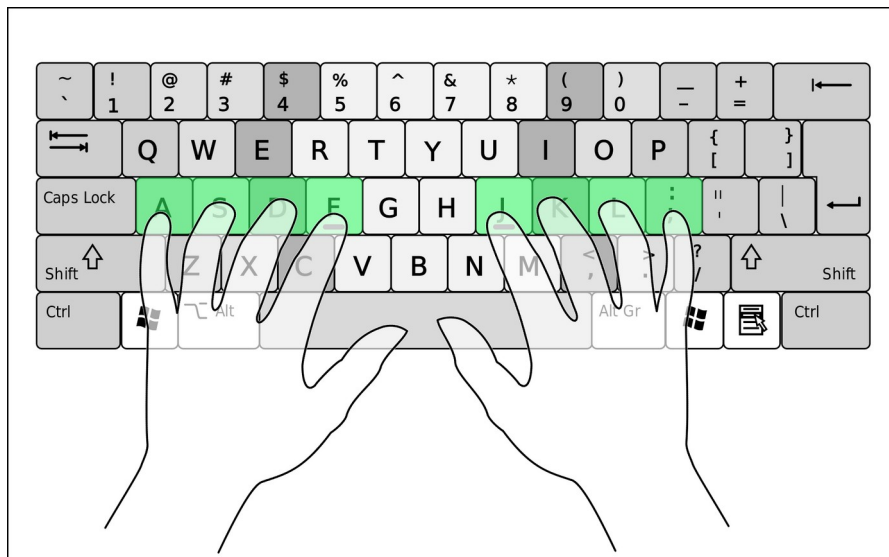
- Nome dell'host (computer remoto)
- Porta
- Nome utente
- Password

SSH è anche il nome di un comando, che permette il collegamento remoto tramite il terminale.

# Touch typing

Una delle tipiche caratteristiche dell'hacker è quella di scrivere molto velocemente alla tastiera :-)

Il *touch typing* è la tecnica che permette di scrivere senza guardare la tastiera.

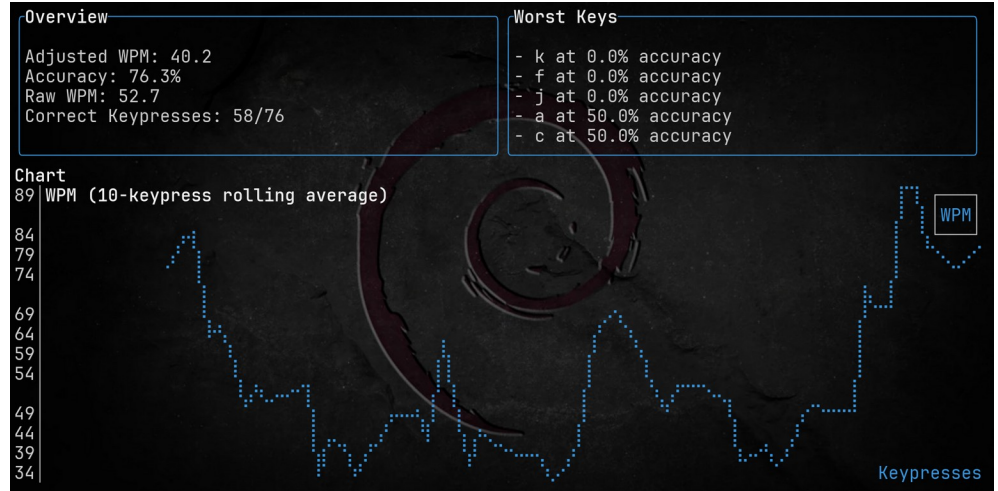


<https://thetypingcat.com/typing-courses>

# Touch typing

Comando:

ttyper



```
ttyper -l /italiano -w 10
```