

## Un valido programma per esercitarsi in telegrafia in Linux

Come in qualsiasi attività umana, per ottenere dei buoni risultati, è necessario esercitarsi. L'esercizio diventa fondamentale anche per migliorare le proprie prestazioni: così come un atleta ha bisogno di esercizio per migliorare, anche un telegrafista necessita di esercitare il proprio orecchio a velocità sempre maggiori.

Un tempo si usavano le cassette: ottimo sistema che, con l'avvento dei moderni computer, è diventato oltremodo desueto. Oggi esistono ottimi programmi per pare pratica: recentemente, essendone alla ricerca, mi sono imbattuto nell'ottimo CWCP, ovviamente per il mondo Linux.

Ho già avuto modo di parlare su queste pagine di questo incredibile sistema operativo, in occasione di un piccolo seminario tenuto a beneficio dei miei amici della Sezione ARI di Lecce: si tratta di un sistema libero, privo di licenze, dove è possibile studiare, disassemblare e modificare i programmi a disposizione dei quali, sempre, è disponibile il sorgente (si tratta, infatti, di programmi Open Source).

Il programma CWCP è disponibile per tutte le distribuzioni Linux, cambiando solo le modalità di installazione; in questa breve disamina, proporrò l'installazione del software su uno dei sistemi più usati: Linux Ubuntu. Il metodo vale per tutte le distribuzioni che da essa derivano (ad esempio Linux Mint che uso) mentre la configurazione del programma “cwcp” vale per qualsiasi distribuzione.

### Cosa permette il programma

Il programma è estremamente leggero: può essere usato direttamente in console (la “*console*” è la parte di Linux che somiglia al vecchio DOS ma, a differenza di questo, è estremamente più potente), senza l'ausilio di una interfaccia grafica (il “*Desktop Manager*” per intenderci, cioè quella videata che permette di avviare i programmi in finestra: molto comoda ma anche “*pesante*”). Avviando il programma nella Shell, avremo la videata visibile in figura 1.

Come si vede, CWCP è abbastanza completo: selezionando le funzionalità nel menù a sinistra, è possibile richiedere l'esecuzione di gruppi di lettere, gruppi di numeri, caratteri alfanumerici, gruppi con tutti i caratteri (lettere, numeri e segni di interpunzione), parole inglesi scelte in modo casuale da un db interno, abbreviazioni del CW e gruppi di caratteri “consimili”, usati per migliorare la loro comprensibilità man mano che la velocità sale. Infatti, a chi non è capitato di confondere una B con un 6 e viceversa?

Ovviamente, in tutto questo, non manca la digitazione da tastiera e, tramite il menù in basso, è possibile modificare i parametri operativi come la velocità (in “WPM” word per minute, parole al minuto, cioè quante volte trasmetto PARIS in un minuto alla velocità impostata), il tono della nota, il volume, lo spazio in punti tra le parole (GAP) e la durata del singolo esercizio.

Ovviamente, avrete notato che... manca la generazione di parole in una lingua diversa dall'inglese! In realtà questo non è un problema in quanto possiamo fare alcune modifiche al file di configurazione ed ottenere la generazione di parole scelte estratte da un testo e la generazione di nominativi casuali scelti dal... database dei nominativi.

Per ottenere tutto questo, in modo che l'articolo sia anche didatticamente valido per tutti coloro che si avvicinano a Linux, procederò per piccoli passi, partendo dall'installazione per arrivare alla configurazione di CWCP, passando per una piccola fase di programmazione (in un ottimo linguaggio di programmazione: il Perl) che ci permetterà di estrarre ed ordinare le parole da un testo scelto tra le migliaia di libri a

disposizione nel *Mare Magnum* di Internet.

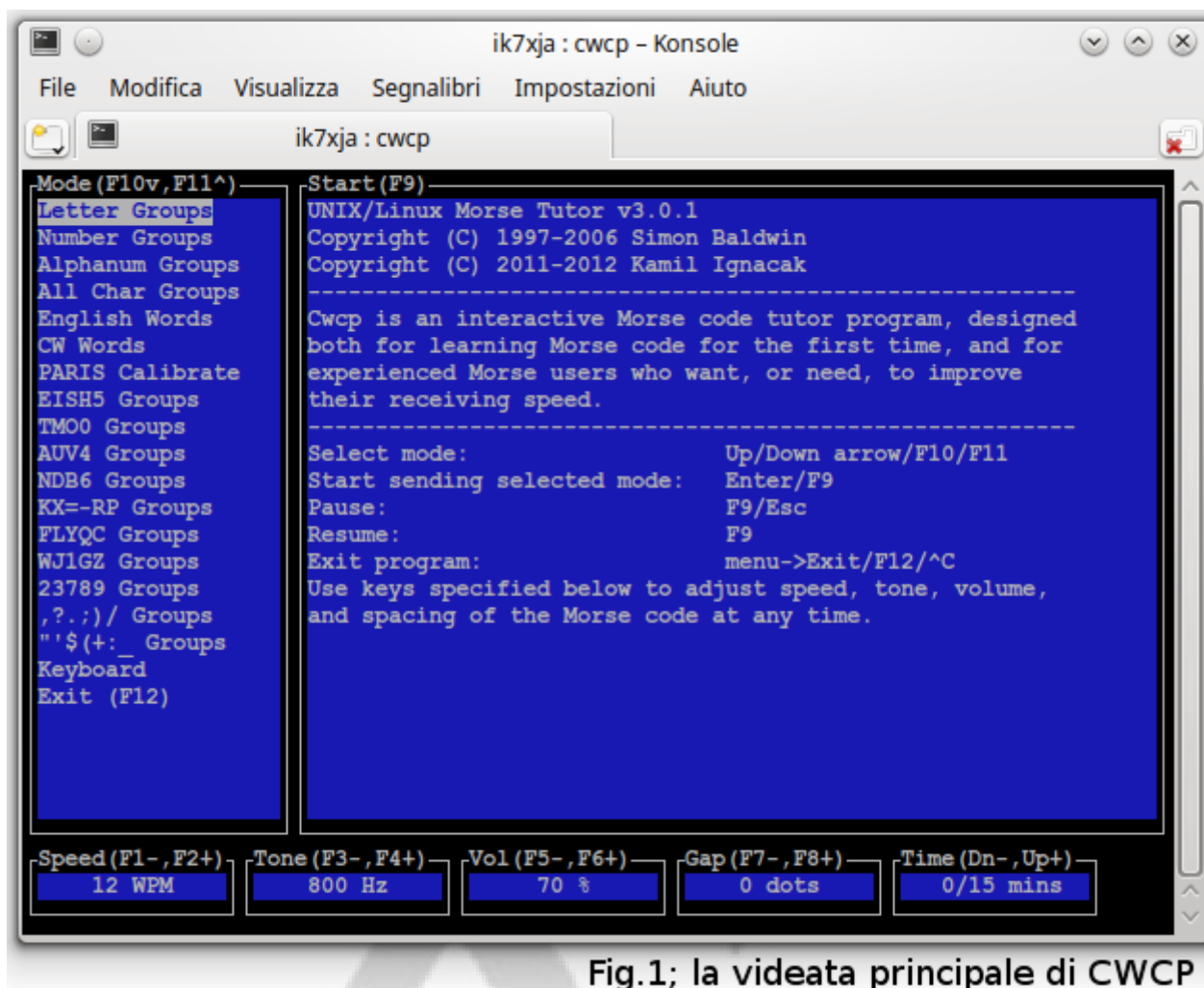


Fig.1; la videata principale di CWCP

## Installazione di CWCP

Usando la distribuzione di Linux Ubuntu o una sua derivata, l'installazione è piuttosto semplice ed il sistema è in grado di risolvere da solo le dipendenze (cioè le librerie software mancanti che devono essere installate e presenti per consentire il corretto funzionamento dei programmi).

Per l'installazione, basta accedere alla Shell (la console, quell'ambiente che somiglia al vecchio DOS) e scrivere i comandi che seguono.

Prima di tutto, verifichiamo che davvero non esista il programma:

```
prompt ~> cwcp
Il programma "cwcp" non è attualmente installato. È possibile installarlo
digitando:
sudo apt-get install cwcp
```

Quindi, il sistema è abbastanza facile: il programma non esiste e viene fornito anche un consiglio di come installarlo; digitiamo, dunque, il comando consigliato:

```
prompt ~> sudo apt-get install cwcp
[sudo] password for user:
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
```

```
Lettura informazioni sullo stato... Fatto
I seguenti pacchetti saranno inoltre installati:
  libcw3
I seguenti pacchetti NUOVI saranno installati:
  cwcp libcw3
0 aggiornati, 2 installati, 0 da rimuovere e 231 non aggiornati.
È necessario scaricare 81,4 kB di archivi.
Dopo quest'operazione, verranno occupati 259 kB di spazio su disco.
Continuare? [S/n] s
Scaricamento di:1 http://ubuntu.mirror.garr.it/mirrors/ubuntu-archive/
trusty/universe libcw3 amd64 3.0.2-1 [35,4 kB]
Scaricamento di:2 http://ubuntu.mirror.garr.it/mirrors/ubuntu-archive/
trusty/universe cwcp amd64 3.0.2-1 [46,0 kB]
Recuperati 81,4 kB in 0s (122 kB/s)
Preconfigurazione dei pacchetti in corso
Selezionato il pacchetto libcw3 non precedentemente selezionato.
(Lettura del database... 220487 file e directory attualmente installati.)
Preparativi per estrarre .../libcw3_3.0.2-1_amd64.deb...
Estrazione di libcw3 (3.0.2-1)...
Selezionato il pacchetto cwcp non precedentemente selezionato.
Preparativi per estrarre .../cwcp_3.0.2-1_amd64.deb...
Estrazione di cwcp (3.0.2-1)...
Elaborazione dei trigger per man-db (2.6.7.1-1ubuntu1)...
Configurazione di libcw3 (3.0.2-1)...
Configurazione di cwcp (3.0.2-1)...
Elaborazione dei trigger per libc-bin (2.19-0ubuntu6.14)...
```

Come si vede, vengono installate le librerie mancanti (“libcw3”) per soddisfare le dipendenze ed il programma richiesto, scaricando il tutto dal mirror (cioè sito internet con i file della distribuzione) più vicino, configurato sul nostro sistema. A questo punto, CWCP è pronto e può essere lanciato digitando, nella shell, il comando che ho indicato all'inizio... È possibile adattarlo per le proprie esigenze, modificando la velocità (WPM, parole al minuto, intendendo per parola il termine “PARIS” usato come metro), la frequenza della nota, il volume e la durata dell'esercizio (c'è un piccolo bug nei tempi: ma non ci importa, in quanto imposteremo i tempi di esecuzione dallo script di avvio).

## Come personalizzare il menu a sinistra

Come anticipato, il programma sarà avviato mediante uno script, in modo da caricare un nostro file di configurazione; per crearne uno da modificare, lanciamo il programma con il seguente parametro:

```
prompt > cwcp -F mio_file_config.txt
```

Il programma sarà avviato (e potremo anche chiuderlo) ma, contestualmente, sarà salvato un file di testo che contiene la configurazione originale, chiamato, proprio, “mio\_file\_config.txt”; esso somiglia molto ai file “.INI” del mondo Windows ed ogni voce di menù è indicata da un testo tra parentesi quadre. Ad esempio:

```
[ Letter Groups ]

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Questo significa che saranno riprodotti i caratteri indicati, cioè solo le lettere dell'alfabeto; parimenti, i

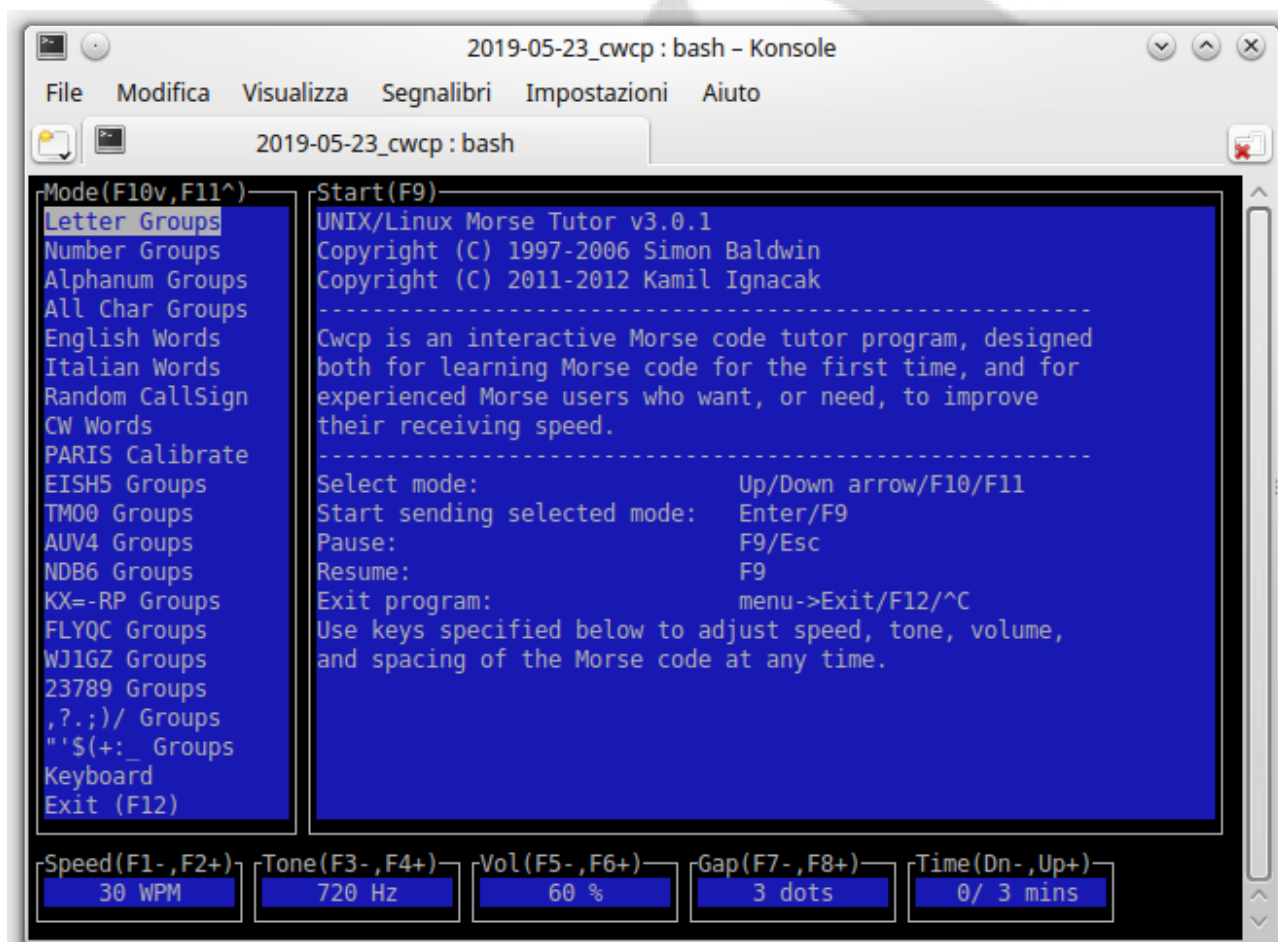
numeri, saranno nel punto successivo:

```
[ Number Groups ]
0 1 2 3 4 5 6 7 8 9
```

Facile vero? Quindi, capito il funzionamento, da bravi italiani amanti della propria lingua, potremmo cominciare con aggiungere delle parole nella nostra lingua, creando una apposita voce di menu:

```
[ Italian Words]
ALL AVO BEN CHE CHI CIO COL CON COS CUI DAI DAL DAR DEH
AMOR ANCH ANCO ANDO ANNI ANNO ANZI APRI ARIA ARTE ATTO AVEA
...
```

Che potremmo mettere dopo il menù che preferiamo; le parole saranno scelte casualmente: più ne mettiamo, meglio sarà. Per caricare il nuovo file di configurazione salvato, basterà ripetere il comando ma, questa volta, con il parametro “-f” (notare che ora è minuscolo). Otterremo la seguente videata:



**Fig. 2: la videata modificata con le nuove voci di menu**

Per lanciare il programma con le impostazioni che preferisco e con i parametri già configurati (velocità, frequenza della nota, gap e durata) ho salvato il file “cw.sh” che contiene i seguenti due righi:

```
#!/bin/sh
cwcp -g 3 -t 720 -w 30 -v 60 -T 3 -f ./mio_file_config.txt
```

Dove:

- `#!/bin/sh`  
indica l'interprete da usare (in questo caso, la shell)
- cwcp, il programma
- `-g 3` il GAP di tre punti (rende più comprensibili i caratteri generati, diminuendo, di fatto la velocità); quando avremo preso dimestichezza, lo porteremo, pian piano, verso lo zero.
- `-t 720`, la frequenza della nota, in hertz
- `-w 30`, la velocità in “parole al minuto” da trasmettere (in questo caso, 30)
- `-v 60`, la perchentuale (60%) del volume del nostro PC
- `-T 3`, la durata di ciascun esercizio in minuti
- `-f file.config`, il nostro file di configurazione personalizzato

È questo è tutto: siamo pronti per iniziare: lanciamo il file “.sh” nel modo seguente:

```
prompt > sh ./cw.sh
```

o, se il file è stato reso eseguibile, senza “sh” all'inizio:

```
prompt > ./cw.sh
```

Ma già vedo qualcuno che chiede: “*ma dobbiamo scrivere le parole italiane ed i nominativi a mano nel file di configurazione?*”. Solo se vogliamo... Nei prossimi paragrafi spiegherò come usare un testo qualsiasi per generare le parole ed ordinarle per lunghezza in modo da aggiungerle, con il copia / incolla nel nostro file di configurazione.

## Estraiamo le parole con un semplice script in Perl

Il Perl (acronimo di “*Practical Extraction and Reporting Language*” cioè linguaggio per pratica estrazione e report) è un linguaggio di scripting potente e facile da usare, nato proprio per lavorare sui testi. Il suo creatore, Larry Wall, è, prima di essere un informatico, un linguista ed il suo linguaggio, al netto di qualche carattere usato come divisore, è molto vicino all'inglese parlato. Il Perl esiste anche per il mondo Windows ma è in Linux che può esprimersi al massimo, potendo contare su una base enorme di moduli pronti per qualsiasi uso ed una perfetta integrazione con il sistema operativo.

La prima cosa da fare è procurarsi un testo abbastanza lungo; io ho scelto il file TXT del primo capitolo del romanzo “Il Castello di Otranto”, versione italiana di un romanzo gotico del 1700 ormai disponibile gratuitamente in rete. Il file del primo capitolo, in puro testo, viene passato come parametro al seguente programma, che riporto con tutti i commenti, in modo che possiate riprodurlo sui vostri PC (script salvato in “`estrai_parole.pl`”):

```
#!/usr/bin/perl
# Estrae le parole uniche da un testo
# Autore: Franco, IK7XJA
# recupero la variabile con il nome del file:
my $file = $ARGV[0] ;
# se il file è valido ed esiste:
```

```

if ($file) {
  if (-e $file) {
    # inizializzo l'array che conterrà il testo
    undef my @righe ;
    # apro il file
    open(IN, "< $file") ;
    @righe = <IN>;
    close(IN);
    # pulizia dei caratteri e sostituzione accenti
    undef my %Parole ;
    my $Max_length = 0 ;
    foreach my $R(@righe) {
      # pongo ogni parola in minuscolo
      $R = lc($R) ;
      # elimino i caratteri speciali e le
      # punteggiature usando espressioni regolari
      $R =~ s/à/a/g ;
      $R =~ s/è/e/g ;
      $R =~ s/é/e/g ;
      $R =~ s/ì/i/g ;
      $R =~ s/ò/o/g ;
      $R =~ s/ù/u/g ;
      $R =~ s/\.//g ;
      $R =~ s/,//g ;
      $R =~ s/;//g ;
      $R =~ s/://g ;
      $R =~ s/\\!//g ;
      $R =~ s/\\?//g ;
      $R =~ s/\\' / /g ;
      $R =~ s/\\\"//g ;
      $R =~ s/\\n//g ;
      $R =~ s/\\r//g ;
      # Divido ogni rigo in parole
      my @Words = split / +/, $R ;
      foreach my $W(@Words) {
        # accumulo le parole solo se non già prese
        # e ne calcolo la lunghezza per ordinarle
        # inoltre, prendo solo quelle con più di due caratteri
        if (length($W)>2) {
          $W = uc($W) ;
          if (!defined($Parole{$W})) {
            $Parole{$W} = 1 ;
            if (length($W)>$Max_length) {
              $Max_length = length($W) ;
            }
          }
        }
      }
    }
  }
}

```

```

# ordino le parole per lunghezza
undef my %Lunghezze ;
foreach my $P(keys %Parole) {
    my $Lun = sprintf("%02d", length($P)) ;
    push(@{$Lunghezze{$Lun}}, $P) ;
}
# in output, ho le parole del testo ordinate
my $Rigo = '' ;
foreach my $L(sort keys %Lunghezze) {
    foreach my $Parola(sort @{$Lunghezze{$L}}) {
        $Rigo .= "$Parola " ;
        if (length($Rigo)>55) {
            print " $Rigo\n" ;
            $Rigo = '' ;
        }
    }
}
} else {
    print "Il file indicato non esiste\n\n" ;
}
} else {
    print "Indicare un testo\n\n" ;
}
}
exit ;

```

Una volta elaborato, il file sarà stampato sullo schermo; per fare in modo che il file sia salvato su un file di testo, basta lanciare lo script come segue:

```
prompt> perl trova_parole.pl capitolo_primo.txt > parole_italiane.txt
```

dove:

“perl” è l'interprete che stiamo usando per far funzionare lo script

“trova\_parole.pl” il nostro file

“capitolo\_primo.txt” il primo capitolo del testo prescelto

“> parole\_italiane.txt” il reindirizzamento dell'output sul nostro file finale

Ora, avendo il file finale, potremmo fare un copia / incolla nel nostro file di configurazione.

Facile vero? Per avere altre parole, magari più attinenti alla radio, basterà cambiare il testo di partenza...

## Nominativi scelti casualmente

Ottimo per impraticarsi è l'ascolto di nominativi generati casualmente; avrei potuto inventare un “generatore” di callsigns ma ho preferito reperire in rete un intero Db e partire da quello. La scelta è ricaduta sull'ottimo programma di Log del collega IN3VRR che usavo prima del passaggio a Linux ed a CQRLog; il programma, disponibile in <http://www.mdxc.org/in3vrr/> contiene una buona base dati di nominativi aggiornata al 2011. La base dati è in DB3 e, per leggerla, installeremo il piccolo programma “dbview” che di solito, non è installato... infatti, provando a lanciare il programma:

```
prompt > dbview
```

Il programma "dbview" non è attualmente installato. È possibile installarlo digitando:

```
sudo apt-get install dbview
```

Anche in questo caso, il sistema riconosce un programma esistente non installato e consiglia il modo di installarlo: l'installazione con il comando suggerito è facile e veloce.

Ora che abbiamo quanto ci serve, scarichiamo il file "world15.zip" dal sito del LogVRR; decompattandolo, abbiamo due file (WORLD.BOK e WORLD.NSX: ci serve solo il primo). Possiamo verificare l'accesso all'archivio con il comando:

```
prompt > dbview -b -t -d '|' WORLD.BOK
```

dove:

dbview è il nostro comando

-b sfoglia l'archivio

-t toglie gli spazi ai record

-d usa il delimitatore "|" (=pipe) per i campi di ogni record

avremo una lista, lunghissima, come questa (dove ho filtrato per i nominativi come il mio):

```
IK7XJA|FRANCO|CASARANO,LE,,||ITALY||LE||JN90BA|ITALY|||
IK7XJC|VINCENZO|BARI,BA,,||ITALY||BA||JN81KC|ITALY|||
IK7XJD|CLAUDIO|GALLIPOLI,LE,,||ITALY||LE||JN80XB|ITALY|||
IK7XJE|UGO F.|FOGGIA,FG,,||ITALY||FG||JN71SL|ITALY|||
IK7XJF|GIOVANNI|MODUGNO,BA,,||ITALY||BA||JN81JB|ITALY|||
```

Come si vede, il primo campo è il nominativo che ci interessa; ora non resta che fare un piccolo script (ovviamente in Perl) per un certo numero di nominativi (non tutti: l'archivio ne contiene quasi 2 milioni!) e creare un output di veri nominativi casuali. Ecco, di seguito, lo script che potrete copiare sul vostro PC. Lo script accetta come parametro il numero di nominativi da estrarre e, siccome ci sono moltissimi nominativi americani, ho limitato la loro presenza ad un quarto del totale degli estratti (script salvato in "nominativi\_casuali.pl"):

```
#!/usr/bin/perl
#
# Genera nominativi casuali dal Callbook in DB3
# Creato da Franco, IK7XJA
# comando dbview: dbview -b -t -d '|' WORLD.BOK
#
# Recupero il parametro passato per il numero di
# nominativi da estrarre; se non esiste, vale 50
my $Quanti_nominativi = $ARGV[0] ;
if (!$Quanti_nominativi) {
    $Quanti_nominativi = 50 ;
}
# Imposto il max dei nominativi USA ad ¼ del totale
my $Max_Nominativi_USA = $Quanti_nominativi / 4 ;
# Definisco la variabile del file del Db (deve stare nella directory)
my $Db = 'WORLD.BOK' ;
```

```

# Definisco il comando di DbView
my $Dbview = qq{/usr/bin/dbview -b -t -d '|'} ;

# Definisco array per i risultati
undef my @Archivio ;

# Lancio il comando in Shell
my $Comando = "$Dbview $Db" ;
my $Res = `$Comando` ;

# Ed assegno il risultato ottenuto all'array
@Archivio = split /\n/, $Res ;
$Res = '' ;

# Comincio l'estrazione dei nominativi casuali
# inizializzando i contatori
my $Max = @Archivio ;
undef my %Generati ;
undef my @Nominativi_estratti ;
my $Quanti_Nominativi_USA = 0 ;
my $Estratti = 0 ;
while ($Estratti < $Quanti_nominativi) {
    my $random_number = int(rand($Max));
    if (!defined($Generati{$random_number})) {
        my $Call = (split /\|/, $Archivio[$random_number])[0] ;
        $Generati{$random_number} = 1 ;
        # conto i nominativi che cominciano per W, N e K (=USA)
        if (($Call =~ /^N/) || ($Call =~ /^K/) || ($Call =~ /^W/)) {
            $Quanti_Nominativi_USA++ ;
            if ($Quanti_Nominativi_USA > $Max_Nominativi_USA) {
                next ;
            } else {
                $Estratti++ ;
                push(@Nominativi_estratti, $Call) ;
            }
        } else {
            $Estratti++ ;
            push(@Nominativi_estratti, $Call) ;
        }
    }
}

# Preparo righe da 55 caratteri al massimo
my $Rigo = '' ;
foreach my $C(@Nominativi_estratti) {
    $Rigo .= "$C " ;
    if (length($Rigo)>55) {
        print " $Rigo\n" ;
        $Rigo = '' ;
    }
}
exit ;

```

Se non ci sono errori di battitura, lanciando lo script nel modo che segue:

```
prompt > perl nominativi_casuali.pl 50
```

Otterremo poco meno di 50 nominati (non viene stampato l'ultimo rigo in quanto ha meno di 55 caratteri):

```
AE7YZ KD4RIZ VU3VRQ W0QWF WB0TNC KD6FGW VA7GRD EA4BVC EA1DJJ  
F1SDE PU1WCP DK6QA KC6FNX N9NGX M6CUB ON7FM HA7VW YH2ADE  
W6ND DL2RBL KK4YYP KJ6YCA DC3APU KO4WB DS2HIQ KI4DEN UA1OGG  
VE3NGH KC7ITC S57ONE IK8OJH VU2QYZ GM0TKC PY2BXN DD9DH IK5GUM  
G0VSP DL8NCU ED5R ZZ1TGH M6BBT S56RDI LW2DTQ IW3HBF HL3HNA
```

Ora sarà sufficiente creare una voce di menu tra parentesi quadre ed inserire i nominativi estratti per completare il nostro file di configurazione personalizzato.

Buona esercitazione!

Se ci sono dubbi, perplessità o consigli non esitate a contattarmi: i dati sono su QRZ.COM.

## Annotazioni

Queste pagine sono state scritte da Franco, IK7XJA e rese disponibili sul sito internet:

- <http://www.ik7xja.it>

nella sezione del sito dedicata al software.