NAME

KeyBoard Wanderer (KBW) − Generates words by walking on a specified keyboard

SYNOPSIS

kbw [**OPTIONS...**]

DESCRIPTION

kbw generates words starting with a specified characters by moving along neighbour keys on a specified keyboard

OPTIONS

-a, --arrangement
path for a keboard configuration file. See **KEYBOARD CONFIGURATION FILE** below.

-d, --dryrun
only count the generated words

-i, --infinite
when the generation is completed causes the program to sleep until a signal is delivered that either terminates the process or causes the invocation of a signal-catching function.

-k, --keys
list of initial main keys. See **LIST OF KEYS** below.

-m, --min
minimum word length to generate

-M, --max
maximum word length to generate

-l, --logfile
log file path. See **LOGFILE** below.

-s, --stop
integer value (> 0) representing a timeout. When the timeout expires a SIGALRM is sent to the process. This option is useful when the **-i**, **--infinite** is set, otherwise the kbw process should be terminated manually.

-w, --restart
to specify a starting string for the generation. The last generated string of a previous run can be used, if the same configuration is used the execution will continue from that point.

USAGE

LOGFILE

The **-l**, **--logfile** option is mandatory and identifies a path to a logfile. This file will contain information on the run and the handled signals. Handled signals are **SIGSEGV**, **SIGTERM**, **SIGINT**, and, **SIGALRM** which is also used in case a timeout is installed (section **TIMEOUT**). When **kbw** is compiled in debug mode (make debug) the log file also shows a message every 500’000’000 generated words (or less at the generation completion).

KEYBOARD CONFIGURATION FILE

The option **-a, --arrangment** is used to supply a keyboard configuration file which defines the se of **keys,** and the **shift variants** for each key, as well as their set of neighbours. The configuration file **must** use an 8-bit character encoding, such as the **ASCII** or **ISO-8859** family. The output will follow the same encoding. The configuration file **must** also adhere to the following structure:

an arbitrary long header composed of either empty lines or lines starting with the character **#** is treated as comment and skipped, the actual line count starts after the header. The header is not mandatory and can be omitted.

Line_1:

N number of main keys (shift variants are not main keys)

Line_2...Line_N+1:

keys in the format **-c<sv>** where **-** is a fixed dash character which precede the key definition, **c,** is the base character for the key, **<sv>** the optional list of **shift variant** characters associated with the base character **c**. The dash character at the beginning of the line is important to distinguish between an empty line and a line defining the space key without **shift variants**.

For example, the line **-1!** means that the main key has associated the character **1**, and there is only one **shift variants** which is the character **!**.

With multiple **shift variants** we can define the line **-+*]}** where **+** is the base character and ***]}** its three **shift variants**.

Line_N+2:

an empty line

Line_N+3...Line_2N+2:

the list of neighbours, represented as main characters, for each key in the format **c:<list_of_main_characters>** for example for distance-1 main key **g** a possible line can be **g:ftyhvb**.

LIST OF KEYS (option -k)

Note that the list of keys passed with the option **-k** should adhere with the encoding used for the configuration file (see **KEYBOARD CONFIGURATION FILE**) otherwise the result may be inconsistent. Pay attention to the encoding automatically used by the Terminal or CLI used to run Keyboard Wanderer, since special characters (like accented letters) are often encoded with multy-byte UTF-8 encoding. As specified in the section **KEYBOARD CONFIGURATION FILE** multi-byte encoding is not allowed in configuration file, as well as in the list of characters passed for this option.

EXAMPLES

KEYBOARD CONFIGURATION FILE

```
---------------------------8<---------------------------
# Test keyboard
#
# full set of base keys to use with -k:
# "1234567890qwertyuiopasdfghjkl’zxcvbnm.  ,"

40
-1!_
-2"#
-3\
-4$@
-5%<
-6ˆ>
-7&[
-8*]
-9({
-0)}
-qQ
-wW
```

```
-eE
-rR
-tT
-yY
-uU
-iI+
-oO-
-pP=
-aA
-sS
-dD
-fF
-gG
-hH
-jJ
-kK
-lL;
-’˜:
-zZ
-xX
-cC
-vV
-bB
-nN
-mM
-.?
-
-,/

1:qw2
2:1we3
3:2er4
4:3rt5
5:4ty6
6:5yu7
7:6ui8
8:7io9
9:8op0
0:9p
q:1wa
w:q12eas
e:w23rsd
r:e34tdf
t:r45yfg
y:t56ugh
u:y67ihj
i:u78ojk
o:i89pkl
p:o90l’
a:qwsz
s:awedzx
d:serfxc
f:drtgcv
g:ftyhvb
```

```
h:gyujbn
j:huiknm
k:jiolm.
l:kop’.
’:pl
z:asx
x:zsdc
c:xdfv
v:cfgb
b:vghn
n:bhjm ,
m:njk. ,
.:mkl,
 :xcvbnm,
,: nm.
-------------------------->8-------------------------
```

LAUNCH

The following example runs the generator with only the **abcd** initial characters, sets min length 1 and max length 3, writing the output to stdout

```
./kbw -a test_keyboard.kbwp -k "abcd" -m 1 -M 3 -l /tmp/logfile.log
```

COPYRIGHT