



Exposé sur DejaGNU

Sommaire

Présentation	1
Comment l'obtenir	1
Comment obtenir la documentation	1
Comment ça marche	1
Remarque	5

Présentation

DejaGnu est un système qui permet d'automatiser le test d'autres programmes, une étape importante si l'on souhaite maintenir un haut niveau de qualité au cours de la vie d'un logiciel. Le paquet *DejaGnu* contient un ensemble d'outils pour tester d'autres programmes.

Avec *DejaGnu*, il existe d'autres frameworks de tests, à savoir : JUnit, C++Unit.

Comment l'obtenir?

Sous la distribution Debian, avec la commande *apt*

```
apt -get dejagnu
```

Sous la distribution Mandrake 10, avec le Manager de Paquetage on peut avoir la version *dejagnu-1.4.3* qui, en parallèle va être installé avec Expect (version ???).

Sous Windows, il existe Cygwin qui est une copie de Unix avec toutes ses commandes standards et son environnement

Sur le site www-ftp.lip6.fr/pub/cygwin/release/dejagnu/ il est possible de télécharger certaines versions de *DejaGnu* pour Linux.

Comment obtenir la documentation?

Tous les sites qui parlent de *DejaGnu*, font référence aux sites suivants pour avoir une documentation complète :

<http://www.gnu.org/software/dejagnu/>

<http://www.gnu.org/software/dejagnu/manual/>

Toute la documentation actuelle sur *DejaGnu* est en Anglais si jamais quelqu'un retrouve plus tard une documentation en Français, faite toujours signe.

Comment ça marche ?

Une fois *DejaGnu* installé :

Testez votre installation

Créer un répertoire vide

```
UserName :$ mkdir NomRep
```

Ce mettre dans ce répertoire crée

```
UserName : $ cd NomRep
```

Maintenant il est possible de tester le programme principal de *DejaGnu*, qui est *Runtest*

```
UserName ~/NomRep$ runtest
WARNING: Couldn't find the global config file.
WARNING: No tool specified Test
Run By UserName on Wed Dec 8 21:07:03 2004 Native configuration is i586-pc-
linux-gnu
=== tests ===
Schedule of variations: unix
Running target unix Using /usr/share/dejagnu/baseboards/unix.exp as board
description file for target.
Using /usr/share/dejagnu/config/unix.exp as generic interface file for
target.
ERROR: Couldn't find tool config file for unix.
=== Summary ===
```

Cette commande crée deux (2) fichiers dans le répertoire courant : *testrun.sum* et *testrun.log* (fichiers contenant les résultats des testes - en faite, ce qui s'affiche dans le shell). Ces fichiers n'ont aucune importance pour l'instant ; nous vous invitons donc à les supprimer.

```
~/NomRep$ rm testrun.sum testrun.log
```

Cela prouve à suffisance que *DejaGnu* est bel et bien installé sur votre distribution ; pour avoir les différentes options qui vont avec *runtest*, faites un man de cette commande.

```
~/NomRep$ man runtest
```

Comme l'on sait que *DejaGnu* fonctionne normalement, il faut mettre dans le répertoire de travail certains fichiers (**configure.in**, **Makefile.am**) et un répertoire (**testsuite**) qui vont accompagner le fichier à tester (ici le fichier **calc.c** - sur le chemin `/usr/share/doc/dejagnu/examples/calc/`).

```
~/NomRep$ cp -r /usr/share/doc/dejagnu/examples/calc/\
{configure.in,Makefile.am,calc.c,testsuite} .
```

Le fichier **configure.in** contient :

```
dnl Process this file with autoconf to produce a configure script.
AC_PREREQ(2.5)
```

```

AC_INIT(calc.c)
AM_CONFIG_HEADER(calc.h)
AM_INIT_AUTOMAKE(calc, 1.1)

AC_PROG_CC
AC_PROG_INSTALL
#
# Look for various header files
#
AC_CHECK_HEADERS(stdlib.h)

#
# Look for various functions
#
AC_CHECK_FUNC(strcmp)

AM_MAINTAINER_MODE

#
# Output Makefile with substitutions
AC_SUBST(CC)
AC_OUTPUT(Makefile)

```

Le fichier **Makefile.am** contient :

```

# Copyright (C) 1992 - 2001 Free Software Foundation, Inc.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 1, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

AUTOMAKE_OPTIONS = dejagnu

noinst_PROGRAMS = calc
calc_SOURCES = calc.c

RUNTESTDEFAULTFLAGS = --tool calc CALC=`pwd`/calc --srcdir
$$srcdir/testsuite

```

Le fichier **calc.c** (fichier à tester) contient :

```

#include "calc.h"

#ifdef HAVE_STDLIB_H
#include <stdlib.h>
#endif

```

```

#include <ctype.h>
#include <stdio.h>

static int words();

#define NWORD 10
#define SIZE 100
#define VERSION "1.1"

int main()
{
    char line[SIZE];
    int nword;
    char *words[NWORD];

    while(printf("calc: "), fflush(stdout), fgets(line,SIZE,stdin) != NULL) {
        if((nword = split(line,words,NWORD)) == 0) continue;
        if(strcmp(words[0],"add") == 0) {
            if(nword != 3) {
                printf("Usage: add #1 #2\n");
            } else {
                printf("%d",atoi(words[1]) + atoi(words[2]));
            }
        } else if(strcmp(words[0],"multiply") == 0) {
            if(nword != 3) {
                printf("Usage: multiply #1 #2\n");
            } else {
                int i1 = atoi(words[1]);
                if(i1 == 2) i1 = 3; /* this is a bug */
                printf("%d",i1*atoi(words[2]));
            }
        } else if(strcmp(words[0],"quit") == 0) {
            break;
        } else if(strcmp(words[0],"version") == 0) {
            printf("Version: %s",VERSION);
        } else {
            printf("Unknown command: %s",words[0]);
        }
        printf("\n");
    }

    return(0);

split(line,words,nword)
char *line;
char **words;
int nword; /* number of elements in words */
{
    int i;

    while(isspace(*line)) line++;
    if(*line == '\0') return(0);

    for(i = 0;i < nword;i++) {
        words[i] = line;
        while(*line != '\0' && !isspace(*line)) line++;
    }
}

```

```

    if(*line == '\0') break;
    *line++ = '\0';
    while(isspace(*line)) line++;
}

return(i);
}

```

Le répertoire **testsuite** contient deux répertoires (**calc.test** et **Config**) :

Le répertoire **calc.test** avec pour fichier **calc.exp** (un fichier Expect) qui contient :

```

# Copyright (C) 1997 - 2001 Free Software Foundation, Inc.

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

# Please email any bugs, comments, and/or additions to this file to:
# bug-dejagnu@gnu.org

set timeout 3
#
# expectations that clean up in case of error. Note that '$test' is
# a purely local variable.
#
# The first of these is used to match any bad responses, and resynchronise
# things by finding a prompt. The second is a timeout error, and shouldn't
# ever be triggered.
#
expect_after {
    -re "\[^\n\r\]*$prompt$" {
        fail "$test (bad match)"
        if { $verbose > 0 } {
            regexp "\.*\r\n(\[^\r\n\]+)(\[\r\n\])+$prompt$" \
                $expect_out(buffer) "" output
            send_user "\tUnmatched output: \"$output\"\n"
        }
    }
    timeout {
        fail "$test (timeout)"
    }
}
#
# Here are the tests
#

```

```

set test "version"
send "version\n"
expect {
    -re "Version:.*$prompt$" { pass "version" }
}

set test add1
send "add 3 4\n"
expect {
    -re "7+.*$prompt$" { pass "$test" }
}

set test add2
send "add 1 2 3\n"
expect {
    -re "Usage: add #1 #2.*$prompt$" { pass "$test" }
}

set test multiply1
send "multiply 3 4\n"
expect {
    -re "12.*$prompt$" { pass "$test" }
}

set test multiply2
send "multiply 2 4\n"
expect {
    -re "8.*$prompt$" { pass "$test" }
}

set test multiply3
send "multiply 1 2 3\n"
expect {
    -re "Usage: multiply #1 #2.*$prompt$" { pass "$test" }
}

```

Dans ce fichier **calc.exp** nous remarquons les différents tests que DejaGnu effectuera sur le fichier **calc.c**

Le répertoire **Config** avec pour fichier **unix.exp** qui contient :

```

# Copyright (C) 1997 - 2001 Free Software Foundation, Inc.

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

```

```

# Please email any bugs, comments, and/or additions to this file to:
# bug-dejagnu@gnu.org

if ![info exists prompt] then {
    set prompt "calc: "
}
#
# calc_version -- extract and print the version number of calc
#

proc calc_version {} {
    global CALC
    global prompt
    set tmp [exec echo "version" | $CALC]
    regexp "$prompt *([^\n]*)\n.*" $tmp tmp version
    clone_output "[which $CALC] version $version\n"
}
#
# calc_load -- loads the program
#
proc calc_load { arg } {
    #
}

#
# calc_exit -- quit and cleanup
#
proc calc_exit {} {
    send "quit\n"
}

#
# calc_start -- start calc running
#
proc calc_start {} {
    global CALC
    global prompt
    global spawn_id
    global verbose

    if { $verbose > 1 } {
        send_user "starting $CALC\n"
    }
    spawn $CALC
    expect {
        -re "No such file.*" { perror "Can't start $CALC"; exit 1 }
        -re "$prompt$" { }
        timeout { perror "Failed to spawn $CALC (timeout)"; exit 1 }
    }
}

calc_start

```

Et une fois que tout cela est fait vous pouvez commencer à créer les outils qui seront utilisé par DejaGnu. Mais cela se fait avec une suite de commandes que voici :

```

:~/NomRep$aclocal

```

Ceci génère le fichier `aclocal.m4` qui est une collection de macro dont le fichier `configure.in` à besoin.

```

:~/NomRep $autoconf
:~/NomRep $autoheader

```

Cette commande génère le fichier `calc.h.in`

```

:~/NomRep $automake --add-missing

```

Permet d'installer certains fichiers dont le programme à besoin pour fonctionner.

```

:~/NomRep $ ./configure

```

Permet d'adapter le programme `calc.c` à l'environnement dans lequel il se trouve, en recherchant par exemple si `gcc` y est ou pas ; s'il n'y est pas, il l'installe.

```

:~/NomRep $make
gcc -DHAVE_CONFIG_H -I. -I. -I. -g -O2 -c calc.c
gcc -g -O2 -o calc calc.o

```

Permet de compiler le programme.

Maintenant, faire un teste du programme.

```

:~/NomRep $make site.exp

```

Génère automatiquement le fichier `site.exp` utile pour lancer le teste de DejaGnu

```

:~/NomRep $make check

```

Lance le test sur le programme `calc.c`

De plus il y a création des fichiers `calc.sum` et `calc.log` comme au début

Contenu de `calc.sum`

```

Test Run By UserName on Thu Dec 9 01:30:05 2004
Native configuration is @host@

```

```

=== calc tests ===

```

Schedule of variations:

```

unix

```

Running target unix

Running `./testsuite/calc.test/calc.exp ...`

PASS: version

PASS: add1

PASS: add2

PASS: multiply1

FAIL: multiply2 (bad match)

PASS: multiply3

```

=== calc Summary ===

```

```

# of expected passes      5

```

```

# of unexpected failures  1

```

```

/home/UserName/NomRep/calc version Version: 1.1

```

Contenu de `calc.log`

```

Test Run By UserName on Thu Dec 9 01:30:05 2004

```

```

Native configuration is @host@

      === calc tests ===

Schedule of variations:
  unix

Running target unix
Using /usr/share/dejagnu/baseboards/unix.exp as board description file for
target.
Using /usr/share/dejagnu/config/unix.exp as generic interface file for
target.
Using ./testsuite/config/unix.exp as tool-and-target-specific interface
file.
calc: Running ./testsuite/calc.test/calc.exp ...
version
Version: 1.1
calc: PASS: version
add 3 4
7
calc: PASS: add1
add 1 2 3
Usage: add #1 #2

calc: PASS: add2
multiply 3 4
12
calc: PASS: multiply1
multiply 2 4
12
calc: FAIL: multiply2 (bad match)
multiply 1 2 3
Usage: multiply #1 #2

calc: PASS: multiply3
testcase ./testsuite/calc.test/calc.exp completed in 0 seconds

      === calc Summary ===

# of expected passes      5
# of unexpected failures   1
/home/UserName/NomRep/calc version Version: 1.1

runtest completed at Thu Dec  9 01:30:06 2004

```

Dans ces deux fichiers, nous remarquons les résultats des testes effectués sur le programme **calc.c**. Ces résultats obéissent à la norme POSIX¹ sur les FrameWords.

Lors de l'exécution du **make check**, l'on peut remarquer une ligne contenant le mot **FAIL**, qui signifie que le programme testé renvoie un résultat erroné.

¹ Cf <http://www.gnu.org/software/dejagnu/manual/x47.html>

Remarque

Je n'ai pu tester DejaGnu qu'avec cet exemple basique. Ce qui fait que les questions comme :

- DejaGnu est-il capable d'effectuer des testes aléatoires (des testes qu'il aura généré tout seul), sans utiliser ceux fait par le programmeurs sous Expect ?
- Un teste de programmation graphique serait-il possible sous DejaGnu ?

Je ne saurais y répondre.

Je suis sûr que cet outil ne se limite pas seulement à ce genre de teste basique que j'ai fait pour pouvoir savoir comment il marche ; plus de recherche sur cet outil nous permettra de savoir jusqu'où il peut aller.