

Bearbeiter: Christoph Lesser; cless@informatik.tu-cottbus.de; 22.05.2004

expect/dejagnu

Herkunft: Tcl/Tk – Sun Microsystems
expect – Don Libes
dejagnu – weiß nicht

Zielsprachen: Tcl bzw. alle Binaries verschiedener Quellsprachen

Plattform: Unix-Derivate; Windows-Systeme

Lizenzstatus: Tcl/Tk – Public Domain;
unter Windows verschiedene kostenpflichtige Versionen
expect/dejagnu – OpenSource

Kurzbeschreibung der Funktionalität

Abstract

Die ToolsCommandLanguage (Tcl) ist eine unter Unix-Systemen verbreitete Skriptsprache für die ein grafisches Toolkit (Tk) existiert. Mit dieser Skriptsprache können Programme auf Kommandozeilenebene gesteuert werden.

Expect ist ein AddOn für diese Sprache, mit der Ausgaben des gesteuerten Programmes gelesen, gefiltert und weiterverarbeitet werden können.

Dejagnu erweitert das expect-AddOn um Methoden zum Regressionstesten nach der POSIX 1003.3-Norm.

Beschreibung

Mit Hilfe des expect/dejagnu-Frameworks können Binaries getestet werden. Diese werden über die Kommandozeile gesteuert. Es handelt sich also um einen BlackBox-Test. Expect erweitert die Tcl um vier Befehle: `spawn` (erzeugt einen neuen Prozess; d.h. ein Programm wird gestartet), `expect` (liest und filtert Ausgaben des Programms von der Kommandozeile und ermöglicht eine verzweigte Skriptsteuerung), `send` (übergibt Daten über die Kommandozeile an den zu testenden Prozess) und `interact` (entkoppelt den Prozess vom Skript und gibt die Kontrolle an den Benutzer ab).

Mit Hilfe dieser vier Befehle findet ein Dialog des zu testenden Prozesses mit dem Testskript statt.

Für eine geeignete Ausgabe der Testergebnisse wurde expect um vier weitere – in der dejagnu festgelegten – Befehle erweitert: `<proc>_load` (lädt und initialisiert das Testskript), `<proc>_start` (beginnt mit der Abarbeitung der Testskripte, die in einem Unterverzeichnis `<proc.test>` liegen und auf `'exp'` enden), `<proc>_exit` (beendet den Testlauf) und `<proc>_version` (liefert die Version des Programms bzw. des Testskriptes).

Durch eine Anpassung der Testskripte kann das Testprotokoll für Menschen oder für Maschinen optimiert werden. Eine Ausgabe in verschiedenen textbasierten Formaten ist so auch möglich (Bsp: PlainASCII, CSV, HTML, XML, eigene Formate)

Das Framework ermöglicht ein automatisiertes Testen der öffentlichen Programmschnittstelle und eignet sich gut für Regressionstests.

Erfahrungen im Umgang

Installation

Ich habe die Installation nur unter Windows getestet. Unter <http://bmrc.berkeley.edu/people/chaffee/expectnt.html> die entsprechende Version des Frameworks herunterladen und das Setup starten. Es wird Tcl/Tk, expect und der graphische Interpreter wish installiert. Die dejagnu-Funktionalität ist nicht enthalten kann aber anhand der Beispiele des Vortrags nachgebildet werden.

Stabilität

Die Testumgebung lief stabil.

Performanz

Die Performanz des Testes orientiert sich im Wesentlichen an der des zu testenden Programms.

Handhabbarkeit

Kenntnisse der Tcl-Syntax erforderlich, da sowohl das expect-AddOn, als auch die Steuerskripte darauf aufsetzen. Ausgaben können mit Hilfe der Tcl formatiert werden und wahlweise für den Menschen oder die Maschine optimiert werden. Insgesamt ist die Handhabung als einfach einzustufen, wenn Kenntnisse im Schreiben von Skripten bzw. Programmen vorhanden sind.

Einarbeitungsaufwand

Wenn die Tcl-Syntax bekannt ist, keiner. Für das expect/dejagnu-Framwork kommen nur vier neue einfach anzuwendende (aber mächtige) Befehle hinzu.

Oberfläche

Kommandoueile (ASCII), aber auch eine graphische Oberfläche mit Hilfe des Toolkits möglich. Für die Aufbereitung eventuell angelegter Testprotokolle (vgl. Handhabbarkeit) können externe Werkzeuge hinzugezogen werden.

Sonstiges

Der Test grafischer Oberflächen des Tcl/Tk-Frameworks ist eingeschränkt möglich. Dazu muss jedoch die interne Struktur des Skriptes/Programmes bekannt sein. (GlassBox-Test zur Laufzeit!). Änderungen an Variablen ziehen Änderungen am Testskript nach sich.