ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

“Ανάπτυξη και διανομή ενός έργου Ελεύθερου Λογισμικού”

ΒΑΣΙΛΑΚΟΣ ΓΕΩΡΓΙΟΣ

Επιβλέπων: Σάββας Ηλίας

ΛΑΡΙΣΑ 2010

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Λάρισα …/…../2010

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. .....
2. .....
3. .....

Σύνοψη

Στη σημερινή εποχή, την εποχή της πληροφορίας, όλο και περισσότερες δραστηριότητές μας βασίζονται στη χρήση Η/Υ και του λογισμικού που το συνοδεύει. Ένα εναλλακτικό μοντέλο ανάπτυξης και χρήσης λογισμικού που βασίζεται στην ελεύθερη διάθεση του πηγαίου κώδικα είναι το Ελεύθερο Λογισμικό.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η παρουσίαση του τρόπου ανάπτυξης και διανομής ενός έργου (project) Ελεύθερου Λογισμικού. Πιο συγκεκριμένα, θα αναπτυχθεί ένα πρόγραμμα εορτολογίου στη γλώσσα προγραμματισμού C με χρήση της εργαλειοθήκης GTK+ για τη δημιουργία της γραφικής διεπαφής. Θα γίνει ανάλυση των εργαλείων που χρησιμοποιήθηκαν για τη μεταγλώττιση του πηγαίου κώδικα και της διαδικασίας διεθνοποίησης του έργου. Ακόμη, θα παρουσιαστεί η διαδικασία για τη δημιουργία Windows installer και πακέτων για τις διανομές Linux Debian και Gentoo ώστε να είναι εύκολος ο διαμοιρασμός και η εγκατάσταση του προγράμματος από τους χρήστες.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου και τους φίλους μου για την συμπαράσταση και την κατανόηση που έδειξαν κατά την διάρκεια της ενασχόλησης μου με την πτυχιακή εργασία αλλά και καθ’ όλη τη διάρκεια της φοιτητικής μου πορείας . Ακόμη, θα ήθελα να ευχαριστήσω τους καθηγητές του τμήματος για την υποστήριξη και τις γνώσεις που αποκόμισα κατά τη διάρκεια των σπουδών. Τέλος, θα ήθελα να ευχαριστήσω την ομάδα Linux του ΤΕΙ Λάρισας για την παροχή σημαντικής βοήθειας και πληροφοριών για την εκπόνηση της πτυχιακής εργασίας.

Με εκτίμηση Βασιλάκος Γεώργιος

Αφιερωμένη στους γονείς μου για την υπομονή, τη στήριξη και την εμπιστοσύνη που μου έδειξαν, καθ' όλη την διάρκεια της φοίτησής μου

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1º

Ελεύθερο Λογισμικό

Ελεύθερο Λογισμικό είναι το λογισμικό που ο καθένας μπορεί ελεύθερα να χρησιμοποιεί, να αντιγράφει, να διανέμει και να τροποποιεί ανάλογα με τις ανάγκες του. Είναι ένα εναλλακτικό μοντέλο ανάπτυξης και χρήσης λογισμικού που βασίζεται στην ελεύθερη διάθεση του πηγαίου κώδικα, το οποίο παρέχει τη δυνατότητα αλλαγών ή βελτιώσεων ώστε να καλύπτονται οι ανάγκες αυτού που το χρησιμοποιεί.

Με βάση αυτή τη φιλοσοφία δημιουργήθηκε μια μεγάλη κοινότητα χρηστών και προγραμματιστών, οι οποίοι συνεργάζονται για τη συνεχή βελτίωση του λογισμικού, παρέχοντας γνώσεις και εργασία. Σήμερα λειτουργεί ένα παγκόσμιο ανοικτό δίκτυο προγραμματιστών, οι οποίοι παράλληλα αναπτύσσουν και διορθώνουν τον κώδικα των προγραμμάτων, κυκλοφορώντας ταχύτατα νέες βελτιωμένες εκδόσεις λογισμικού. Με αυτές τις συνεχείς βελτιώσεις και την αυξημένη πλέον φιλικότητα προς το χρήστη, το Ελεύθερο Λογισμικό κερδίζει διαρκώς νέους φίλους παγκοσμίως. Στην εκπαίδευση, στη δημόσια διοίκηση και στις επιχειρήσεις, οι ενδιαφερόμενοι ενημερώνονται και αποκτούν ιδιαίτερα ελκυστικά εργαλεία, αξιόπιστα, σταθερά στη λειτουργία, και απαλλαγμένα από τα σημαντικά κόστη απόκτησης και συνεχούς αναβάθμισης που απαιτούν τα κλειστά λογισμικά. Έτσι, πλέον όλο και πιο πολλοί πόροι διατίθενται στην τεχνική υποστήριξη με σημαντικά οφέλη για την τοπική και εθνική οικονομία.

1.1 Ιστορική αναδρομή

Στη δεκαετία του 1950, του 1960 και 1970 θεωρούταν φυσιολογικό για τους χρήστες υπολογιστών να έχουν τις ελευθερίες που παρέχονται σήμερα από το ελεύθερο λογισμικό. Το λογισμικό εκείνη την εποχή ήταν ελεύθερα διαθέσιμο σε άτομα που χρησιμοποιούσαν ηλεκτρονικούς υπολογιστές και οι κατασκευαστές hardware ήταν ευτυχείς που οι άνθρωποι ανέπτυσσαν λογισμικό το οποίο έκανε το υλικό τους χρήσιμο. Είχαν δημιουργηθεί ακόμη και οργανώσεις χρηστών και προμηθευτών για να διευκολυνθεί η ανταλλαγή λογισμικού, όπως για παράδειγμα το SHARE user group. Στα τέλη της δεκαετίας του 1960 όμως μια αλλαγή θα ερχόταν καθώς το κόστος για την ανάπτυξη και συντήρηση λογισμικού από τους κατασκευαστές hardware αυξάνονταν ραγδαία. Μια εκκολαπτόμενη και διαρκώς αναπτυσσόμενη βιομηχανία λογισμικού ήταν ανταγωνιστική με τα πλήρη συστήματα λογισμικού του κατασκευαστή του υλικού (του οποίου το κόστος συμπεριλαμβανόταν στο υλικό). Το αγορασμένο hardware έπρεπε να παρέχει υποστήριξη λογισμικού, ωστόσο οι κατασκευαστές δεν είχαν τα έσοδα για το λογισμικό, καθώς ορισμένοι πελάτες καλύπτονταν από το δωρεάν λογισμικό και δεν ήθελαν το κόστος των προϊόντων hardware να συμπεριλαμβάνει και αυτό του λογισμικού.

Το 1969 η αποκλειστική διάθεση λογισμικού μαζί με το hardware κρίνεται από την κυβέρνηση των ΗΠΑ ως βλαπτική για τον ανταγωνισμό. Έτσι, ενώ ένα μεγάλο πλήθος λογισμικού θα συνέχιζε να διανέμεται δωρεάν, άρχισε να αναπτύσσεται και λογισμικό το οποίο ήταν μόνο προς πώληση. Στη δεκαετία του 1970 και στις αρχές της δεκαετίας του 1980 η βιομηχανία λογισμικού ξεκίνησε με τη βοήθεια τεχνικών μέτρων, όπως η διάθεση μόνο των εκτελέσιμων μορφών των προγραμμάτων, να περιορίζει τη δυνατότητα των χρηστών υπολογιστή να μελετήσουν και να τροποποιήσουν το λογισμικό. Το 1980 ο νόμος περί πνευματικών δικαιωμάτων επεκτάθηκε και στα προγράμματα ηλεκτρονικών υπολογιστών. Οι εταιρίες ανάπτυξης λογισμικού άρχισαν να χρεώνουν άδειες χρήσης, διαφημίζοντας το λογισμικό τους ως "πρόγραμμα προϊόν" και επιβάλλοντας νομικούς περιορισμούς εφόσον πλέον το λογισμικό θεωρούταν περιουσιακό στοιχείο.

Το 1983 ο Richard Stallman, απογοητευμένος από την αλλαγή νοοτροπίας στον κλάδο της πληροφορικής και των χρηστών, ανακοίνωσε το GNU project. Το όνομα GNU είναι αναδρομικό ακρωνύμιο του "GNU's Not Unix" (GNU δεν είναι Unix). Το GNU project είχε ως σκοπό τη δημιουργία ενός ολοκληρωμένου λειτουργικού συστήματος τύπου Unix το οποίο θα ήταν εξολοκλήρου ελεύθερο λογισμικό. Η ανάπτυξη λογισμικού για το λειτουργικό σύστημα GNU ξεκίνησε τον Ιανουάριο του 1984 και τον Οκτώβριο του 1985 ιδρύθηκε το Ίδρυμα Ελεύθερου Λογισμικού (FSF). Το Ίδρυμα Ελεύθερου Λογισμικού είναι ένας μη-κερδοσκοπικός οργανισμός που ιδρύθηκε για την υποστήριξη του κινήματος ελεύθερου λογισμικού και ειδικότερα του GNU project. Ακόμη, ο Richard Stallman ανέπτυξε τον ορισμό του ελεύθερου λογισμικού και την έννοια "copyleft", με σκοπό να εξασφαλίσει την ελευθερία του λογισμικού για όλους. Copyleft είναι το όνομα ενός τύπου αδειών χρήσης για ότι αφορά τα πνευματικά δικαιώματα. Οι πιο δημοφιλείς copyleft άδειες είναι η Γενική Άδεια Δημόσιας Χρήσης GNU (GNU General Public License – GNU GPL) που αφορά το λογισμικό και η GNU Free Documentation License που αφορά την τεκμηρίωση των προγραμμάτων.

1.2 GNU General Public License

Λογότυπο της Γενικής άδειας δημόσιας χρήσης

Η GPL είναι η πρώτη και πιο διαδεδομένη copyleft άδεια χρήσης, που σημαίνει ότι τα παράγωγα της αρχικής δημιουργίας πρέπει να διανέμονται κάτω από τους ίδιους όρους χρήσης. Κάτω από αυτή τη φιλοσοφία, η GPL προσφέρει στον δημιουργό ενός προγράμματος τα διακαιώματα που κατοχυρώνονται από τον ορισμό του ελεύθερου λογισμικού, κάνοντας χρήση του copyleft για να διασφαλίσει ότι αυτές οι ελευθερίες θα παραμείνουν ακόμα και αν υπάρξουν αλλαγές ή προσθήκες. Αυτό γίνεται για να διαφοροποιηθεί από άλλες άδειες ελεύθερου λογισμικού όπως η άδεια BSD.

Το κείμενο της GPL δεν είναι κάτω από την άδεια GPL οπότε δεν επιτρέπονται αλλαγές σε αυτήν. Ωστόσο η αντιγραφή και διανομή της άδειας χρήσης επιτρέπεται όπως αναγράφεται και στο κείμενο της ρητά: "ο αποδέκτης πρέπει να λάβει και ένα αντίγραφο της άδειας χρήσης μαζί με το πρόγραμμα". Σύμφωνα με το FAQ της GPL, οποιοσδήποτε μπορεί να αλλάξει την άδεια όσο η νέα άδεια θα έχει διαφορετικό όνομα, δεν θα περιέχει το όνομα GNU και θα αφαιρεθεί ο πρόλογος της άδειας. Ο πρόλογος μπορεί να χρησιμοποιηθεί σε μια τροποποιημένη άδεια μόνο με την άδεια του Ιδρύματος Ελεύθερου Λογισμικού.

Εκδοση 1η

Η πρώτη έκδοση της Γενικής Άδειας Δημόσιας Χρήσης δημοσιεύτηκε το 1989 με σκοπό να άρει τους περιορισμούς που επέβαλαν τότε οι διανομείς λογισμικού. Το πρώτο πρόβλημα ήταν ότι διανέμονταν μόνο το εκτελέσιμο αρχείο ενός προγράμματος το οποίο όμως δε μπορεί να διαβαστεί και τροποποιηθεί από τους χρήστες. Έτσι η GPLv1 ανέφερε πως αν κάποιος διανέμει το εκτελέσιμο αρχείο ενός προγράμματος, αυτό θα πρέπει να συνοδεύεται και από τον πηγαίο κώδικα του προγράμματος, ο οποίος είναι κατανοητός από τον άνθρωπο, κάτω από την ίδια άδεια χρήσης,

Ο δεύτερος περιορισμός ήταν ότι οι διανομείς λογισμικού μπορούσαν να προσθέσουν επιπλέον περιορισμούς στην άδεια χρήσης, ή να συνδυάσουν προγράμματα με περιορισμούς σε άλλα προγράμματα. Αν γινόταν αυτό, το τελικό πρόγραμμα θα είχε και το σύνολο των περιορισμών αυτών. Για να το αποφύγει αυτό, η GPLv1, ανέφερε ότι οι τροποποιημένες εκδόσεις λογισμικού θα πρέπει να διανέμονται κάτω από την ίδια άδεια. Έτσι, λογισμικό κάτω από την άδεια GPLv1 δε θα

μπορούσε να συνδυαστεί με λογισμικό με περισσότερους περιορισμούς καθώς αυτό θα παρέβαινε την προϋπόθεση το σύνολο του λογισμικού να διατίθεται κάτω από τους όρους της GPLv1.

Έκδοση 2η

Σύμφωνα με τον Richard Stallman, η βασική αλλαγή στη δεύτερη έκδοση της άδειας ήταν η ρήτρα "Ελευθερία ή θάνατος". Το κείμενο αυτής της έκδοσης δηλώνει ρητά ότι αν κάποιος χρήστης δέχεται περιορισμούς στην πρόθεσή του να διανείμει λογισμικό κάτω από την άδεια GPL, δε θα πρέπει να το διανέμει καθόλου.

Ήδη από το 1990 είχε αρχίσει να γίνεται φανερό ότι μια νέα άδεια λιγότερο περιοριστική θα ήταν πολύ χρήσιμη για τη βιβλιοθήκη της C και άλλες βιβλιοθήκες που στην ουσία αντικαθιστούσαν ιδιόκτητες βιβλιοθήκες. Όταν η δεύτερη έκδοση της GPL κυκλοφόρησε τον Ιούνιο του 1991, συνοδεύτηκε από μια επιπλέον άδεια, την LGPL. Ο αριθμός της έκδοσης άλλαξε το 1999 σε έκδοση 2.1 όταν η LGPL μετονομάστηκε σε GNU Lesser General Public License, όνομα το οποίο αντιπροσώπευε καλύτερα τη φιλοσοφία της άδειας.

Έκδοση 3η

Στα τέλη του 2005 το Ίδρυμα Ελεύθερου Λογισμικού ανακοίνωσε ότι βρισκόταν σε εξέλιξη η τρίτη έκδοση της άδειας GPL. Τον Ιανουάριο του 2006 κυκλοφόρησε το πρώτο προσχέδιο της άδειας για να τεθεί σε αξιολόγηση από την κοινότητα. Ενώ η περίοδος αξιολόγησης αρχικά είχε υπολογιστεί στους εννέα με δεκαπέντε μήνες, τελικά είχε διάρκεια δεκαοχτώ μήνες, στο τέλος των οποίων κυκλοφόρησαν τέσσερα προσχέδια. Η επίσημη τρίτη έκδοση της Γενικής Άδειας Δημόσιας Χρήσης κυκλοφόρησε από το Ίδρυμα Ελεύθερου Λογισμικού στις 29 Ιουνίου του 2007.

Σύμφωνα με τον Stallman, οι βασικότερες αλλαγές αφορούσαν τις πατέντες λογισμικού, τη συμβατότητα των αδειών ελεύθερου λογισμικού και τον ορισμό του "Ανοικτού κώδικα". Άλλες αλλαγές αφορούσαν τη διεθνοποίηση της άδειας, πώς θα αντιμετωπίζονται οι παραβιάσεις της άδειας και πώς μπορούν να παρέχονται περισσότερα δικαιώματα από τον κάτοχο του λογισμικού. Άλλη μια σημαντική αλλαγή έδινε το δικαίωμα στο δημιουργό να προσθέτει επιπλέον όρους ή απαιτήσεις σε ό,τι αφορά τη συμμετοχή σε έργα ελεύθερου λογισμικού.

Η διαδικασία της δημόσιας αξιολόγησης της άδειας αυτής συντονίστηκε από το FSF σε συνεργασία με το Software Freedom Law Center, Free Software Foundation Europe και άλλες

ομάδες ελεύθερου λογισμικού. Τα σχόλια των χρηστών συλλέγονταν μέσα από την ιστοσελίδα gplv3.fsf.org και διανέμονταν σε τέσσερις επιτροπές που αποτελούνταν συνολικά από εκατόν τριάντα άτομα, συμπεριλαμβανομένων υποστηρικτών και επικριτών των στόχων του Ιδρύματος Ελεύθερου Λογισμικού. Οι επιτροπές αυτές μελετούσαν τα σχόλια του κοινού και προωθούσαν τα συμπεράσματα στον Stallman προκειμένου να φτάσει στην τελική απόφαση του τι θα περιλαμβάνει η άδεια. Κατά τη διάρκεια της αξιολόγησης δημοσιεύτηκαν 962 σχόλια για το πρώτο προσχέδιο, ενώ μέχρι το τέλος της τα σχόλια ανήλθαν σε 2636.

Το τρίτο προσχέδιο κυκλοφόρησε το Μάιο του 2007 και περιλάμβανε ρήτρες που απαγόρευαν τις πατέντες λογισμικού. Ακόμη, αφαιρέθηκε η παράγραφος περί γεωγραφικών περιορισμών, όπως προτάθηκε και στη δημόσια συζήτηση.

Το τέταρτο και τελευταίο προσχέδιο, ανακοινώθηκε στις 31 Μαΐου 2007. Εισήγαγε τη συμβατότητα με την άδεια χρήσης Apache και προσδιόρισε το ρόλο των εξωτερικών επιχειρήσεων στην ανάπτυξη του ελεύθερου λογισμικού.

Ο στόχος όλων των εκδόσεων της GPL είναι η διασφάλιση ότι καθένας που λαμβάνει ένα αντίγραφο λογισμικού με άδεια GPL είναι ελεύθερος να χρησιμοποιεί αυτό το λογισμικό, να το μετατρέπει και να το διανέμει σε τροποποιημένη ή μη μορφή. Η έκδοση 3 αποδίδεται σε αλλαγές στο νομικό και τεχνικό περιβάλλον του λογισμικού, και ανατρέπει νέες τεχνικές οι οποίες εφαρμόστηκαν για να αφαιρέσουν αυτές τις ελευθερίες από τους χρήστες του λογισμικού.

Αναφορές

http://www.gnu.org/gnu/manifesto.el.html

http://www.gnu.org/gnu/gnu-history.html

http://www.gnu.org/gnu/thegnuproject.html

http://www.fsfe.org/projects/gplv3/gplv3.el.html

http://en.wikipedia.org/wiki/History_of_free_software

http://en.wikipedia.org/wiki/GNU_General_Public_License

ΚΕΦΑΛΑΙΟ 2º

Γραφική διεπαφή

Γραφική διεπαφή ή γραφικό περιβάλλον χρήστη (αγγλικά: Graphical User Interface - GUI) είναι ένα σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής, όπως ο ηλεκτρονικός υπολογιστής, και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Τα περισσότερα σύγχρονα προγράμματα και λειτουργικά συστήματα προσφέρουν στους χρήστες τους κάποια γραφική διεπαφή γιατί αυτός ο τρόπος αλληλεπίδρασης με τον υπολογιστή ταιριάζει αρκετά στην ανθρώπινη εμπειρία και φύση. Σωστά σχεδιασμένα γραφικά προσφέρουν ένα όμορφο, εύχρηστο, λειτουργικό και πολύ παραγωγικό περιβάλλον εργασίας.

Οι σύγχρονες γραφικές διεπαφές χρησιμοποιούν ένα συνδυασμό από τεχνολογίες και συσκευές ώστε να παρέχουν στο χρήστη μια εύκολη πλατφόρμα με την οποία αλληλεπιδρούν. Τα πιο σημαντικά μέρη από τα οποία αποτελείται ένα γραφικό περιβάλλον, τα οποία ονομάζονται widgets, είναι τα παράθυρα, τα εικονίδια, τα μενού και ο δείκτης του ποντικιού.

2.1 Εργαλειοθήκες γραφικής διεπαφής

Για να διευκολυνθεί η διαδικασία της δημιουργίας μιας γραφικής διεπαφής σε κάποια εφαρμογή, αναπτύχθηκαν κάποιες εργαλειοθήκες γραφικής διεπαφής. Μια τέτοια εργαλειοθήκη παρέχει μια μεγάλη ποικιλία από widgets τα οποία μπορεί να χρησιμοποιήσει ο προγραμματιστής μέσω της διασύνδεσης προγραμματισμού εφαρμογών (API) που παρέχει η εργαλειοθήκη. Τα widgets συνήθως προσαρμόζονται ώστε να δίνουν την αίσθηση συνολικής συνοχής μεταξύ των διαφόρων τμημάτων της εφαρμογής και γενικά μεταξύ των εφαρμογών εντός του γραφικού περιβάλλοντος. Ακόμη, πολλές εργαλειοθήκες παρέχουν τη δυνατότητα αλλαγής της εμφάνισης των widgets με τη χρήση θεμάτων.

Μια γραφική διεπαφή σε κάποιο πρόγραμμα συνήθως κατασκευάζεται κλιμακωτά με widgets να προστίθενται πάνω από υπάρχοντα widgets. Για παράδειγμα ένα παράθυρο μπορεί να περιέχει μια λίστα από κουμπιά που περιέχουν από μια εικόνα και ένα κείμενο. Οι πιο δημοφιλής εργαλειοθήκες χρησιμοποιούν ως πρότυπο για την αλληλεπίδρασή τους με το χρήστη, προγραμματισμό οδηγούμενο από γεγονότα (event-driven programming). Αυτό σημαίνει ότι η εργαλειοθήκη εντοπίζει γεγονότα του χρήστη, όπως για παράδειγμα όταν ο χρήστης κάνει κλικ σε ένα κουμπί, και τα διοχετεύει στην εφαρμογή όπου αντιμετωπίζονται ανάλογα.

Στις μέρες μας ένας προγραμματιστής μπορεί να επιλέξει την εργαλειοθήκη που τον βολεύει από μια πολύ μεγάλη λίστα. Η επιλογή αυτή μπορεί να γίνει βάση της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί, της πλατφόρμας στην οποία απευθύνεται το πρόγραμμα, των πόρων

συστήματος που θα καταναλώνονται και της γενικότερης εμφάνισης της εφαρμογής. Οι πιο δημοφιλείς ανοικτού κώδικα εργαλειοθήκες για τη δημιουργία γραφικής διεπαφής είναι η wxWidgets, η QT και η GTK+, ενώ πολύ έδαφος κερδίζουν οι σύγχρονες εργαλειοθήκες που παρέχουν τρισδιάστατη απεικόνιση, όπως η Clutter.


λογότυπο wxWidgets

Η εργαλειοθήκη wxWidgets (πρώην wxWindows) ξεκίνησε το 1992 από τον Julian Smart στο πανεπιστήμιο του Εδιμβούργου. Είναι γραμμένη στη γλώσσα προγραμματισμού C++, διανέμεται κάτω από μια άδεια ελεύθερου λογισμικού και λειτουργεί σε όλα τα λειτουργικά συστήματα. Ακόμη, παρέχει τη διασύνδεση ώστε να μπορεί να χρησιμοποιηθεί στις πιο δημοφιλείς γλώσσες προγραμματισμού όπως C++, Pyhton, Ruby και άλλες. Το μεγάλο πλεονέκτημά της έναντι άλλων εργαλειοθηκών είναι ότι τα widgets εμφανίζονται και συμπεριφέρονται σαν να είναι μέρος του λειτουργικού συστήματος. Δηλαδή αν μια εφαρμογή μεταγλωττιστεί στην πλατφόρμα των Windows θα έχει την εμφάνιση και την αίσθηση μιας εφαρμογής για Windows, ενώ αν μεταγλωττιστεί σε Linux θα έχει την ομοιογένεια με εφαρμογές του Linux.

λογότυπο QT

Η εργαλειοθήκη QT ξεκίνησε να αναπτύσσεται το 1991 από την εταιρία Trolltech, την οποία αγόρασε το 2008 η εταιρία Nokia. Αν και αρχικά δεν ήταν ελεύθερο λογισμικό, από τον Ιούνιο του 2003 η άδεια χρήσης άλλαξε σε GNU GPL πράγμα που την έκανε ακόμη πιο δημοφιλή δημιουργώντας μια πολύ μεγάλη και ενεργή κοινότητα. Όπως και η εργαλειοθήκη wxWidgets, έτσι και η QT μπορεί να χρησιμοποιηθεί από πολλές γλώσσες προγραμματισμού και λειτουργεί σε όλα τα λειτουργικά συστήματα, ενώ επιπλέον μπορεί να χρησιμοποιηθεί σε ενσωματωμένα συστήματα όπως κινητά τηλέφωνα. Ακόμη, προσφέρει τη δυνατότητα για λειτουργίες εκτός γραφικής διεπαφής όπως διαχείριση βάσεων δεδομένων, αρχείων στη γλώσσα XML, νημάτων και άλλα. Στην ευρεία διάδοση αυτής της εργαλειοθήκης συνέβαλε και ένα από τα πιο δημοφιλή γραφικά περιβάλλοντα για το λειτουργικό σύστημα Linux, το KDE, το οποίο την χρησιμοποιεί εξ' ολοκλήρου σε όλες τις εφαρμογές του.

2.2 Η εργαλειοθήκη GIMP (GTK+)

Λογότυπο GTK+

Η εργαλειοθήκη GTK+ είναι μια από τις ισχυρότερες εργαλειοθήκες για τη δημιουργία γραφικής διεπαφής. Είναι γραμμένη στη γλώσσα προγραμματισμού C, ωστόσο παρέχει διασυνδέσεις για τις περισσότερες γλώσσες προγραμματισμού. Αρχικά αναπτύχθηκε για να παρέχει μια όμορφη και εύχρηστη διεπαφή στο GIMP, ένα πανίσχυρο πρόγραμμα επεξεργασίας εικόνων. Όταν οι Peter Mattis και Spencer Kimball άρχισαν να αναπτύσσουν το GIMP το 1995 δεν υπήρχε κάποια ισχυρή και ελεύθερη βιβλιοθήκη που να τους καλύπτει, οπότε ξεκίνησαν να δημιουργούν τη δική τους. Παρ' όλο που η εργαλειοθήκη GIMP, ή GTK, δημιουργήθηκε συγκεκριμένα για τις ανάγκες του GIMP, βρήκε πολύ γρήγορα εφαρμογή σε πολλά παρόμοια έργα. Όταν προστέθηκε η κληρονομικότητα στη βιβλιοθήκη, την έκανε πιο αντικειμενοστραφή και έτσι προστέθηκε το σύμβολο + στο όνομα της βιβλιοθήκης για να αναδείξει αυτή τη διαφορά. Σήμερα η επίσημη ονομασία της είναι GTK+, αν και το σύμβολο + συνήθως δεν προφέρεται.

Η εργαλειοθήκη GTK+ είναι μια βιβλιοθήκη υψηλού επιπέδου, ωστόσο παρέχει και λειτουργίες πολύ χαμηλού επιπέδου. Τέτοιες λειτουργίες είναι η σχεδίαση στην οθόνη και η εμφάνιση κειμένου με χρήση κάποιων επιπλέον βιβλιοθηκών που λειτουργούν πιο κοντά στο σύστημα στο οποίο τρέχουν. Αυτές οι βιβλιοθήκες αναπτύσσονται παράλληλα με τη GTK+ και, παρ' όλο που μπορούν χρησιμοποιηθούν και ξεχωριστά από αυτή, είναι πιο χρήσιμες μέσα στο πλαίσιο της GTK+. Οι κυριότερες από αυτές οι βιβλιοθήκες είναι η GDK, η GdkPixbuf, η Pango και η Glib.

Η βιβλιοθήκη GDK (Graphics Development Kit) είναι υπεύθυνη για την εμφάνιση των συστατικών της γραφικής διεπαφής στην οθόνη. Παρέχει λειτουργίες σχεδίασης στην οθόνη, λήψης πληροφοριών από το ποντίκι, το πληκτρολόγιο και άλλες συσκευές, ακόμη και μεθόδους drag & drop και αντιγραφής – επικόλλησης. Για να παρέχει διαλειτουργικότητα στη GTK+, η βιβλιοθήκη GDK αποτελείται από ένα σύνολο διαφορετικών υποβάθρων, καθένα από τα οποία λειτουργεί σε διαφορετικά συστήματα. Το πιο κοινό υπόβαθρο είναι του X για εμφάνιση γραφικών στο X Window System το οποίο χρησιμοποιείται κατά κόρον σε συστήματα τύπου Unix, και το Win32 για τα Windows.

Η βιβλιοθήκη GdkPixbuf χρησιμοποιείται για τη διαχείριση αρχείων εικόνων. Μπορεί να γράψει και να διαβάσει τους πιο γνωστούς τύπους εικόνων όπως .jpeg, .gif, .bmp και .ico.

Προσφέρει λειτουργίες απλής επεξεργασίας όπως αλλαγή μεγέθους, overlaying και desaturating. Ακόμη, παρέχει απευθείας πρόσβαση στα δεδομένα της εικόνας, προσφέροντας τη δυνατότητα για άμεση επεξεργασία ανά πάσα στιγμή. Η πιο κοινή χρήση της βιβλιοθήκης είναι η φόρτωση μιας εικόνας σαν εικονίδιο εφαρμογής, αν και μπορεί να χρησιμοποιηθεί γενικότερα για την εμφάνιση και διαχείριση εικόνων μέσα στην εφαρμογή.

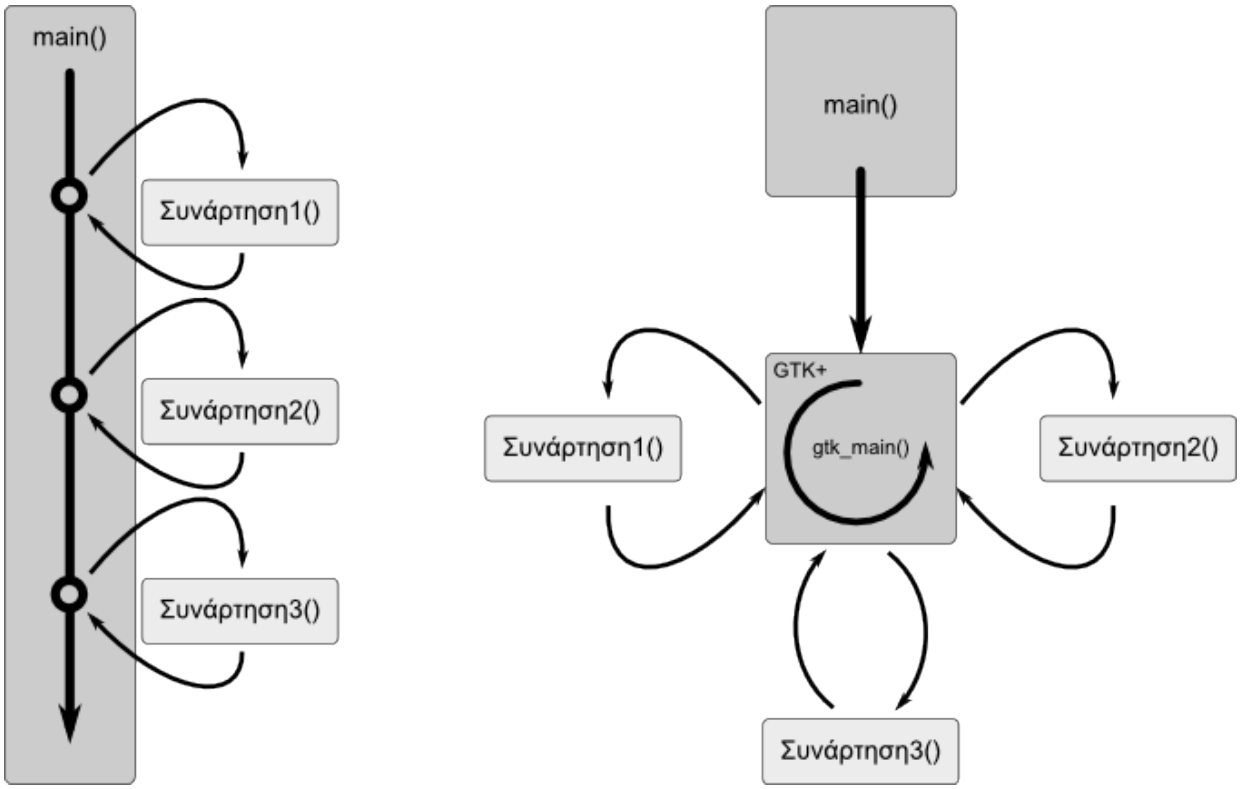
Η βιβλιοθήκη Pango χρησιμοποιείται για την εμφάνιση κειμένου σε οποιαδήποτε γλώσσα, αναπαριστώντας το κείμενο στην κωδικοποίηση UTF-8. Ο προγραμματιστής μπορεί να εφαρμόσει διάφορα χαρακτηριστικά σε διαφορετικά κομμάτια του κειμένου. Τέτοια χαρακτηριστικά είναι η αλλαγή χρώματος, μεγέθους, πλάτους, γραμματοσειράς και άλλων. Στη συνέχεια η Pango θα αναλάβει να εμφανίσει το κείμενο σύμφωνα με τις παραπάνω τροποποιήσεις, λαμβάνοντας υπόψιν θέματα όπως η διεθνοποίηση και οι ρυθμίσεις γραφής.

Οι προγραμματιστές σπάνια κάνουν κατευθείαν κλήσεις σε συναρτήσεις της Pango καθώς οι περισσότερες λειτουργίες μπορούν να χρησιμοποιηθούν μέσα από τα widget της GTK+. Για παράδειγμα, αν μια λεζάντα έχει όνομα "Το κείμενο είναι έντονο" αυτόματα θα εμφανιστεί στο χρήστη σαν "Το κείμενο είναι **έντονο**". Για να είναι ευκολότερη η χρήση των πιο κοινών χαρακτηριστικών παρέχονται ετικέτες (tags) παρόμοιες με αυτές της γλώσσας HTML.

Η βιβλιοθήκη Glib παρέχει τις δομές δεδομένων, τη διαλειτουργικότητα και άλλες χρήσιμες συναρτήσεις. Για να επιτευχθεί η διαλειτουργικότητα ανεξαρτήτως αρχιτεκτονικής και λειτουργικού συστήματος, υπάρχουν ειδικοί τύποι δεδομένων, όπως gchar, gint και άλλοι. Κομμάτι αυτής της βιβλιοθήκης είναι και η συνάρτηση gtk_main την οποία χρησιμοποιούν οι εφαρμογές GTK+ για την αλληλεπίδραση με το χρήστη και τη συνεχή ανανέωση της εμφάνισης της εφαρμογής.

2.3 Ανατομία μιας εφαρμογής GTK+

Η εφαρμογή GTK+ διαφέρει από τα περισσότερα διαδικαστικά προγράμματα της C. Αντί για την καθαρή σειριακή διαδικασία που ακολουθείται σε ένα τυπικό πρόγραμμα, κατά την οποία μπορεί κανείς να προβλέψει τα πάντα από την αρχή μέχρι το τέλος, σε μια εφαρμογή GTK+ υπάρχει μια μεγάλη ομάδα από ασύνδετες συναρτήσεις. Αυτές οι συναρτήσεις συνήθως δεν καλούνται από το ίδιο το πρόγραμμα αλλά ενεργοποιούνται από κάποιο γεγονός το οποίο έχει ορίσει ο προγραμματιστής κατά τη δημιουργία της γραφικής διεπαφής, όπως το πάτημα ενός κουμπιού. Στην εικόνα 2.1 φαίνεται η διαφορά από ένα τυπικό διαδικαστικό πρόγραμμα και από ένα πρόγραμμα με χρήση της GTK+.



Εικόνα 2.1: Το διάγραμμα ροής ενός τυπικού προγράμματος (αριστερά), και ενός με χρήση της GTK+ (δεξιά).

Εάν θέλουμε να δημιουργήσουμε ένα κουμπί το οποίο να αποθηκεύει ένα αρχείο όταν πατηθεί, πρέπει να δημιουργήσουμε τη συνάρτηση και να πληροφορήσουμε τη GTK+ να την καλέσει όταν το κουμπί πατηθεί. Παρόμοια, αν ένα κουμπί χρειάζεται να δημιουργήσει ένα νέο παράθυρο, πρέπει πρώτα να δημιουργήσουμε μια συνάρτηση που θα δημιουργεί το νέο παράθυρο

και έπειτα να πληροφορήσουμε τη GTK+ να την καλέσει όταν το κουμπί πατηθεί. Αυτό που στην ουσία κάνει η συνάρτηση main() είναι να αρχικοποιεί τη GTK+ και τα υπόλοιπα κομμάτια της εφαρμογής και στη συνέχεια δίνει τον έλεγχο στη GTK+, η οποία καλεί τις υπόλοιπες συναρτήσεις της εφαρμογής όπως πρέπει.

Η main() είναι η πρώτη συνάρτηση που καλείται σε ένα πρόγραμμα της C και το πρόγραμμα τερματίζεται όταν τερματιστεί και αυτή η συνάρτηση. Ένα πρόγραμμα με γραφική διεπαφή παραμένει στη συνάρτηση main() μέχρι ο χρήστης να τερματίσει το πρόγραμμα. Αυτό επιτυγχάνεται με μια επαναληπτική main η οποία περιμένει συνεχώς κάποια αλληλεπίδραση από το χρήστη και επανασχεδιάζει τη γραφική διεπαφή όταν αυτό είναι απαραίτητο. Μόνο με δύο κλήσεις συναρτήσεων μπορείτε να γράψετε μια εφαρμογή με γραφική διεπαφή η οποία θα αρχικοποιεί τον εαυτό της και θα δίνει έπειτα τον έλεγχο στη GTK+.

Πριν το κάλεσμα οποιασδήποτε άλλης συνάρτησης του προγράμματος, πρέπει να κληθεί η συνάρτηση gtk_init(). Αυτή η συνάρτηση αρχικοποιεί τη βιβλιοθήκη και προσπελαύνει τα ορίσματα που περάστηκαν στη γραμμή εντολών.

Όλες οι εφαρμογές GTK+ μπορούν να δεχτούν από τη γραμμή εντολών συγκεκριμένα ορίσματα που παρέχονται από τη βιβλιοθήκη της GTK+. Σ' αυτά τα ορίσματα περιλαμβάνονται επιλογές για αποσφαλμάτωση και χαμηλού επιπέδου λειτουργίες για τους έμπειρους χρήστες. Η gtk_init() προσπελαύνει τα ορίσματα και αντιδρά μόνο σε αυτά που κατανοεί, αφαιρώντας τα από τη λίστα των υπόλοιπων ορισμάτων ώστε να μπορούν να προσπελαστούν από το υπόλοιπο πρόγραμμα. Τα ορίσματα της γραμμής εντολών περνούν στη συνάρτηση main() μέσω της μεταβλητής argv και το πλήθος των ορισμάτων μέσω της μεταβλητής argc. Στη γλώσσα προγραμματισμού C, περνώντας ένα δείκτη σε μια συνάρτηση γίνεται κλήση με αναφορά στα δεδομένα που περάστηκαν και έτσι η καλούσα συνάρτηση μπορεί να αλλάξει τα περιεχόμενα του δείκτη. Αν η κλήση στη συνάρτηση γίνει με τιμή, τότε η καλούσα συνάρτηση δεν μπορεί να αλλάξει την τιμή αυτή. Για αυτόν ακριβώς το λόγο η συνάρτηση gtk_init() παίρνει σαν ορίσματα δείκτες των μεταβλητών argv και argc ώστε να μπορεί να αλλάξει τα περιεχόμενά τους, για παράδειγμα gtk_init(&argc, &argv).

Η εφαρμογή παραμένει στην επαναληπτική main της GTK+ καθ' όλη τη διάρκεια ζωής της διαδικασίας. Η επαναληπτική main είναι ένας ατέρμων βρόγχος ο οποίος δέχεται συνεχώς ερεθίσματα από το ποντίκι, το πληκτρολόγιο και άλλες συσκευές, ανανεώνει την οθόνη και διαχειρίζεται πληροφορίες όπως το πάτημα ενός κουμπιού μέσω ενός συστήματος σημάτων και λειτουργιών. Από τη συνάρτηση main() απλά καλούμε τη gtk_main() η οποία δε δέχεται ορίσματα

και θα τερματίσει όταν κληθεί η συνάρτηση gtk_main_quit() οπουδήποτε στον κώδικα του προγράμματος.

Το συντομότερο έγκυρο πρόγραμμα που μπορεί κανείς να γράψει σε GTK+ μπορεί να αποτελείται μόνο από δύο συναρτήσεις:

```
#include <gtk/gtk.h>
int main ( int argc, char *argv[] )
{
   gtk_init(&argc, &argv);
   gtk_main();
   return 0;
}
```

Αυτή η εφαρμογή βέβαια δε θα εκτελεί καμιά λειτουργία, θα παραμείνει κολλημένη στη gtk_main() για πάντα.

Αναφορές

http://en.wikipedia.org/wiki/Graphical_user_interface

http://www.wxwidgets.org

http://en.wikipedia.org/wiki/Qt_(framework)

http://library.gnome.org/devel/gtk/2.16/

http://en.wikipedia.org/wiki/GTK

Open source messaging application development: building and extending Gaim, Sean Egan, Apress, 1 edition (July 4, 2005), ISBN-13: 978-1590594674

ΚΕΦΑΛΑΙΟ 3º

GNU Build System

Το GNU build system, γνωστό και ως Autotools, είναι μια σουίτα από προγραμματιστικά εργαλεία που δημιουργήθηκαν από το GNU project. Αυτά τα εργαλεία σχεδιάστηκαν με σκοπό να βοηθήσουν τη δημιουργία και τη διανομή λογισμικού σε λειτουργικά συστήματα τύπου Unix. Το GNU build system χρησιμοποιείται στα περισσότερα έργα ελεύθερου λογισμικού. Τα εργαλεία από τα οποία αποτελείται το GNU build system είναι κάτω από την άδεια GNU GPL με μερικές εξαιρέσεις που επιτρέπουν τη χρήση του για τη δημιουργία ιδιόκτητου λογισμικού.

Τα προγραμματιστικά εργαλεία που παρέχονται από το GNU build system είναι το Autoconf, το Automake και το Libtool. Συνήθως χρησιμοποιούνται σε συνδυασμό με άλλα εργαλεία του GNU project όπως το GNU make, το GNU gettext, το pkg-config, και το GNU Compiler Collection, γνωστό και ως GCC.

3.1 GNU Compiler Collection

Λογότυπο GCC

Το GNU Compiler Collection, εν συντομία GCC, είναι ένα σύστημα μεταγλώττισης το οποίο αναπτύσσεται από το GNU Project. Ουσιαστικά είναι μια συλλογή από μεταγλωττιστές για τις κυριότερες γλώσσες προγραμματισμού. Αυτή τη στιγμή είναι το προεπιλεγμένο σύστημα μεταγλώττισης στα περισσότερα λειτουργικά συστήματα τύπου Unix, συμπεριλαμβανομένων του GNU/Linux, της οικογένειας των BSD και του Mac OS X. Το GCC έχει μεταφερθεί σε πολλές αρχιτεκτονικές επεξεργαστών, είναι διαθέσιμο για τις περισσότερες embedded πλατφόρμες και χρησιμοποιείται ευρέως ακόμη και σε εμπορικά, ιδιόκτητα λογισμικά. Υπάρχουν ακόμη και εταιρίες με στόχο την παροχή και υποστήριξη της μεταφοράς του GCC σε διάφορες πλατφόρμες, ενώ ακόμη και οι κατασκευαστές chip θεωρούν τη μεταφορά του σχεδόν απαραίτητη για την επιτυχία μιας αρχιτεκτονικής.

Η ανάπτυξη του GCC ξεκίνησε από τον Richard Stallman το 1985 επεκτείνοντας έναν υπάρχοντα μεταγλωττιστή ώστε να μεταγλωττίζει στη γλώσσα προγραμματισμού C. Ωστόσο αυτός ο μεταγλωττιστής λειτουργούσε μόνο με την Pastel, μια μη μεταφέρσιμη διάλεκτο της γλώσσας προγραμματισμού Pascal. Έτσι ο μεταγλωττιστής γράφτηκε από την αρχή στη γλώσσα C από τους Len Tower και τον Stallman, και κυκλοφόρησε σαν ελεύθερο λογισμικό από το GNU Project το 1987. Έως το 1991, η πρώτη έκδοση του GCC είχε φτάσει σε ένα αρκετά ικανοποιητικό επίπεδο

από άποψη σταθερότητας, αλλά κάποιοι αρχιτεκτονικοί περιορισμοί εμπόδιζαν πολλές επιθυμητές βελτιώσεις του, οπότε τότε το Ίδρυμα Ελεύθερου λογισμικού άρχισε να αναπτύσσει τη δεύτερη έκδοσή του.

Καθώς το GCC είναι ελεύθερο λογισμικό, πολλοί προγραμματιστές ήθελαν να εστιάσουν σε άλλες κατευθύνσεις εκτός από τη βελτιστοποίηση του, όπως η λειτουργία με επιπλέον γλώσσες προγραμματισμού εκτός από τη C, και έτσι δημιούργησαν παρακλάδια του μεταγλωττιστή. Ωστόσο πολλά από αυτά τα παρακλάδια αποδείχτηκαν αναποτελεσματικά και δύσχρηστα, με αποτέλεσμα να μην είναι επίσημα αποδεκτά από το GCC project και το Ίδρυμα Ελεύθερου Λογισμικού που το εποπτεύει. Πλέον στην επίσημη έκδοση 4.3 του GCC περιλαμβάνονται διεπαφές για τις γλώσσες προγραμματισμού C (gcc), C++ (g++), Java (gcj), Ada (GNAT), Fortran (gfortran), Objective-C (gobjc) και Objective-C++ (gobjc++). Ακόμη υπάρχουν διαθέσιμες διεπαφές, όχι επισήμως, για τις γλώσσες Modula-2, Modula-3, Pascal (gpc), PL/I, D (gdc), Mercury και VHDL (ghdl). Ακόμη, υποστηρίζονται συνολικά πάνω από 60 αρχιτεκτονικές επεξεργαστών.

Ο τρόπος λειτουργίας του GCC αρχικά είναι να καλεί ένα πρόγραμμα-οδηγό με όνομα gcc, το οποίο ερμηνεύει τα ορίσματα που έχουν έχουν δοθεί από τη γραμμή εντολών. Στη συνέχεια αποφασίζει ποιο μεταγλωττιστή θα χρησιμοποιήσει για το κάθε αρχείο του πηγαίου κώδικα και τον καλεί. Τέλος, εκτελεί τον assembler για τη δημιουργία των αντικειμένων (objects), ενώ συνήθως εκτελεί έπειτα και τον linker ώστε να παραχθεί το τελικό εκτελέσιμο αρχείο. Ο μεταγλωττιστής για την κάθε γλώσσα προγραμματισμού είναι ένα ξεχωριστό πρόγραμμα, το οποίο δέχεται τον πηγαίο κώδικα και παράγει κώδικα σε assembly, αλλά όλοι έχουν μια κοινή εσωτερική δομή. Μια διεπαφή (front-end) σε κάθε γλώσσα προγραμματισμού είναι υπεύθυνη για τη συντακτική ανάλυση του πηγαίου κώδικα στη γλώσσα αυτή και παράγει ένα abstract syntax tree.

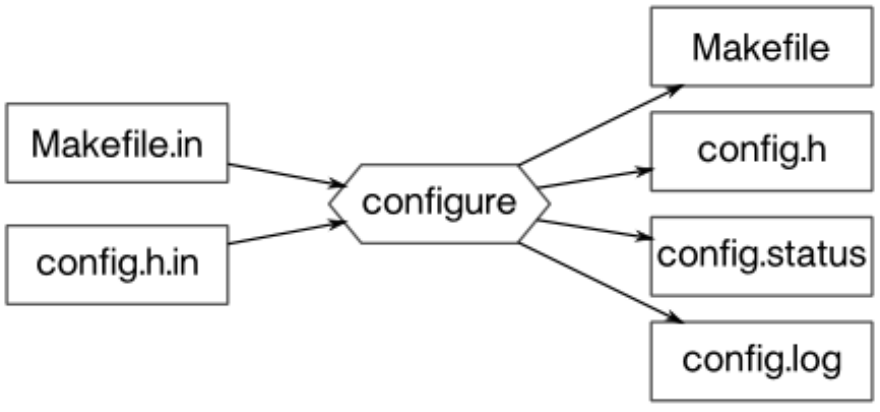
Για τη μεταγλώττιση του προγράμματος Giortes, το οποίο δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας, χρησιμοποιήθηκε η έκδοση 4.3.4 του GCC για το λειτουργικό σύστημα Linux και η έκδοση 3.4.5 για το λειτουργικό σύστημα Windows XP.

3.2 Το εργαλείο GNU Autoconf

Το GNU Autoconf είναι ένα εργαλείο το οποίο παράγει σενάρια ρυθμίσεων (configure scripts) για το πακέτο του λογισμικού, τα οποία το ρυθμίζουν κατάλληλα πριν τη μεταγλώττιση και εγκατάστασή του. Μετά την εκτέλεση μιας σειράς από αυτόματους ελέγχους, ένα configure script δημιουργεί προσαρμοσμένα αρχεία header και makefile από τα πρότυπα που έχουν ήδη δημιουργηθεί. Αυτά τα αρχεία προσαρμόζουν το λογισμικό και τη διαδικασία εγκατάστασης ώστε να ταιριάζει στο σύστημα του χρήστη. Ο κύριος σκοπός του Autoconf είναι να κάνει ευκολότερη τη ζωή του χρήστη και έπειτα του συντηρητή του λογισμικού, διευκολύνοντας τη διαδικασία της μεταγλώττισης και κάνοντας το πρόγραμμα μεταφέρσιμο.

Το βασικό configuration script που παράγει το Autoconf ονομάζεται configure. Όταν εκτελεστεί το configure script δημιουργεί κάποια αρχεία και τα ρυθμίζει κατάλληλα. Αναλυτικότερα, τα αρχεία που δημιουργούνται είναι:

- ένα ή περισσότερα αρχεία Makefile, συνήθως ένα σε κάθε υποφάκελο
- προαιρετικά ένα αρχείο C header, το οποίο περιέχει κάποιους ορισμούς #define
- ένα σενάριο κελύφους (shell script) με όνομα config.status το οποίο όταν εκτελεστεί, δημιουργεί ξανά τα παραπάνω αρχεία
- ένα αρχείο με όνομα config.log το οποίο περιέχει όλες τις πληροφορίες που παράχθηκαν από τους μεταγλωττιστές, ώστε να βοηθήσει στην αποσφαλμάτωση εάν υπάρχει κάποιο λάθος στο configure



Εικόνα 3.1: Το διάγραμμα ροής του configure script. Αριστερά είναι τα αρχεία εισόδου και δεξιά τα αρχεία που παράγονται.

Για τη δημιουργία του configure script, χρειάζεται να γράψουμε ένα αρχείο εισόδου για το autoconf, το οποίο ονομάζεται configure.ac και στη συνέχεια να εκτελέσουμε την εντολή autoconf. Εάν θέλουμε να χρησιμοποιήσουμε και κάποιο αρχείο C header με ορισμούς #define, πρέπει να εκτελέσουμε και την εντολή autoheader. Στο πρόγραμμα Giortes έχει χρησιμοποιηθεί τέτοιο αρχείο κυρίως για να δηλωθεί κατάλληλα η σταθερά GTK_GUI ή NO_GUI η οποία ελέγχεται σε αρκετά σημεία του κώδικα ώστε να αποφασιστεί αν θα γίνει χρήση της γραφικής διεπαφής ή όχι. Ακόμη ορίζονται σταθερές με το όνομα του πακέτου, την έκδοσή του, την ιστοσελίδα του, στοιχεία και τρόποι επικοινωνίας με το συντηρητή του πακέτου.

Το πρόγραμμα autoscan μας βοηθά στη δημιουργία και τη συντήρηση του αρχείου configure.ac. Το autoscan εξετάζει τον πηγαίο μας κώδικα για συνήθη προβλήματα μεταφερσιμότητας και δημιουργεί ένα αρχείο configure.scan. Αυτό το αρχείο μπορεί να χρησιμοποιηθεί σαν βάση για το δικό μας αρχείο configure.ac, ή εάν έχουμε ήδη δημιουργήσει το configure.ac, το ελέγχει αν είναι πλήρες. Όταν γίνεται χρήση του autoscan πρέπει να εξετάζεται χειροκίνητα το αρχείο configure.scan πριν μετονομαστεί σε configure.ac, καθώς συνήθως χρειάζεται κάποιες παραμετροποιήσεις. Μερικές φορές το autoscan ρυθμίζει επίτηδες λάθος το παραγόμενο αρχείο έτσι ώστε το autoconf στη συνέχεια να παράγει μια προειδοποίηση και να αναγκάσει τον προγραμματιστή να ελέγξει το αρχείο χειροκίνητα.

Στο παραπάνω πρότυπο αρχείο configure.scan έχουν γίνει αρκετές τροποποιήσεις και προσθήκες μέχρι να επιτευχθεί στο επιθυμητό αποτέλεσμα και το τελικό αρχείο configure.ac. Τα περιεχόμενα του αρχείου είναι κάποιες κλήσεις σε μακροεντολές οι οποίες θα ελέγχουν αν το σύστημα έχει όλα τα προαπαιτούμενα για να μεταγλωττίσει και να ρυθμίσει το λογισμικό. Μια από τις κυριότερες μακροεντολές που τροποποιήθηκαν και υπάρχουν σε κάθε πακέτο λογισμικού είναι η AC_PREREQ στην οποία ορίζεται η ελάχιστα απαιτούμενη έκδοση του autoconf. Ακόμη, η AC_INIT, η οποία προσπελαύνει τα ορίσματα της γραμμής εντολών και εκτελεί κάποιες αρχικοποιήσεις και επαληθεύσεις, ενώ επιπλέον ορίζει κάποιες πληροφορίες για το πακέτο όπως το όνομα και την έκδοση.

Για να λειτουργήσει το header file που χρειαζόμαστε, προστέθηκε η μακροεντολή AC_CONFIG_HEADERS. Ακόμη, προστέθηκε η AM_GNU_GETTEXT η οποία χρειάζεται ώστε το autoconf να διασφαλίζει ότι το πακέτο ανταποκρίνεται σε ορισμένες από τις ανάγκες του gettext, του εργαλείου που βοηθά στη διεθνοποίηση του έργου. Ακολουθούν κάποιες μακροεντολές AC_CHECK_HEADERS οι οποίες εξετάζουν την ύπαρξη κάποιων header files και μια AC_MSG_CHECKING η οποία ειδοποιεί το χρήστη πως θα γίνει κάποιος έλεγχος. Ο παραπάνω

έλεγχος γίνεται για να διαπιστωθεί το λειτουργικό σύστημα του χρήστη, με χρήση της μεταβλητής host η οποία παρέχεται από το autoconf, και στη συνέχεια ορίζεται κατάλληλα μια δική μας μεταβλητή platform_win32. Έπειτα, εξετάζεται η παραπάνω μεταβλητή και εάν το σύστημα είναι Windows ελέγχονται κάποια επιπλέον header files και ορίζονται κατάλληλα βιβλιοθήκες που θα χρησιμοποιηθούν κατά το link.

Μια ακόμη σημαντική μακροεντολή που προστέθηκε είναι η AC_ARG_ENABLE, η οποία ελέγχει αν κατά την εκτέλεση του configure script ο χρήστης έχει επιλέξει να ενεργοποιήσει κάποια επιπλέον χαρακτηριστικά με χρήση του ορίσματος --enable-χαρακτηριστικό. Στην περίπτωση του προγράμματος Giortes, αυτό το χαρακτηριστικό είναι η χρήση γραφικής διεπαφής και το όνομα του ορίσματος πρέπει να είναι --enable-gtk-gui. Αν έχει ενεργοποιηθεί το παραπάνω χαρακτηριστικό τότε γίνεται έλεγχος για την εργαλειοθήκη GTK+ με τη μακροεντολή AM_PATH_GTK_2_0 και ορίζονται ξανά κατάλληλα οι βιβλιοθήκες ώστε να συμπεριλαμβάνονται αυτές της γραφικής διεπαφής. Επίσης, γίνεται χρήση της μακροεντολής AC_DEFINE η οποία ορίζει τη σταθερά GTK_GUI ή NO_GUI στο header file που ορίζεται από την AC_CONFIG_HEADERS. Ακόμη, χρησιμοποιείται η μακροεντολή AC_CONFIG_FILES η οποία παράγει ένα νέο αρχείο, περιέχοντας τις κατάλληλες τιμές για το παρόν σύστημα, για κάθε αρχείο που έχει σαν όρισμα. Τέλος, υπάρχει η AC_OUTPUT η οποία είναι υπεύθυνη για τη δημιουργία και εκτέλεση του shell script με όνομα config.status. Συνήθως, τα ορίσματα που παίρνει η μακροεντολή AC_CONFIG_FILES είναι τα Makefiles που υπάρχουν σε κάθε φάκελο και τους υποφακέλους του. Η μακροεντολή AC_OUTPUT πρέπει πάντα να είναι η τελευταία του αρχείου configure.ac.

3.3 Το εργαλείο GNU Automake

Το GNU Automake είναι ένα προγραμματιστικό εργαλείο το οποίο παράγει μεταφέρσιμα Makefiles τα οποία χρησιμοποιεί το πρόγραμμα make για τη μεταγλώττιση και εγκατάσταση του πακέτου λογισμικού. Είναι γραμμένο στη γλώσσα προγραμματισμού Perl και συνήθως χρησιμοποιείται σε συνδυασμό με το GNU Autoconf. Το Automake περιέχει την εντολή aclocal, η οποία δημιουργεί ένα βοηθητικό αρχείο aclocal.m4 σύμφωνα με το αρχείο configure.ac, και την automake η οποία δημιουργεί αρχεία Makefile.in από τα αρχεία Makefile.am. Ένα αρχείο Makefile περιέχει κάποιους κανόνες που προσδιορίζουν πώς θα μεταγλωττιστεί και εγκατασταθεί το τελικό πρόγραμμα από τις εξαρτήσεις του.

Στόχος του Automake είναι να επιτρέπει στον προγραμματιστή να γράψει ένα makefile σε γλώσσα ανώτερου επιπέδου, αντί να γράψει όλο το Makefile με το χέρι, δηλώνοντας τα βασικότερα

στοιχεία που χρειάζονται για το πρόγραμμά του. Τέτοια στοιχεία είναι το όνομα του προγράμματος, μια λίστα με τα αρχεία του πηγαίου κώδικα, μια λίστα με ορίσματα της γραμμής εντολών που πρέπει να περαστούν στον compiler (για παράδειγμα, που βρίσκονται κάποια header files) και μια λίστα με ορίσματα της γραμμής εντολών που πρέπει να περαστούν στον linker (για παράδειγμα, ποιες βιβλιοθήκες χρειάζεται το πρόγραμμα και πού βρίσκονται). Με τις παραπάνω πληροφορίες το Automake σε συνδυασμό με το Autoconf, δημιουργεί ένα αρχείο Makefile το οποίο επιτρέπει εύκολα στο χρήστη να μεταγλωττίσει το πρόγραμμα, να το εγκαταστήσει στους κατάλληλους φακέλους, να το απεγκαταστήσει και να δημιουργήσει ένα πακέτο με τον κώδικά του έτοιμο προς διανομή, γνωστό και σαν tarball.

Συνήθως ο προγραμματιστής πρέπει να γράψει ένα αρχείο Makefile.am σε κάθε φάκελο του πακέτου. Στο λογισμικό Giortes η δομή του πακέτου περιέχει τέσσερις φακέλους, τους data, pixmap, po και src.

Ο φάκελος data περιέχει κάποια αρχεία που χρειάζονται για να λειτουργήσει σωστά το πρόγραμμα και κάποια επιπλέον που βοηθούν την ενσωμάτωση του λογισμικού στο σύστημα. Τέτοια αρχεία είναι το database.dat που περιέχει τις γιορτές, το αρχείο συντόμευσης Giortes.desktop και το giortes.1 που περιέχει την τεκμηρίωση της εφαρμογής. Τα αρχεία .desktop χρησιμοποιούνται στο λειτουργικό σύστημα Linux ως ένας ενιαίος τρόπος χρήσης συντομεύσεων ανεξαρτήτως γραφικού περιβάλλοντος και έχουν μια συγκεκριμένη δομή η οποία ορίζεται από τις προδιαγραφές του freedesktop.org. Το αρχείο giortes.1 είναι το man page της εφαρμογής το οποίο θα εμφανίζεται όταν ο χρήστης πληκτρολογήσει την εντολή man giortes. Η κατάληξη .1 δηλώνει ότι πρόκειται για κάποιο γενικό πρόγραμμα. Η δομή αυτού του αρχείου συνήθως περιλαμβάνει το όνομα της εντολής και μια μικρή περίληψή της σε μια γραμμή, τη σύνοψη της εντολής, την περιγραφή των λειτουργιών της, κάποια παραδείγματα και πηγές για επιπλέον πληροφορίες.

Στο φάκελο pixmap είναι τα εικονίδια giortes64.png και giortes32.xpm που θα χρησιμοποιεί η εφαρμογή και το γραφικό περιβάλλον. Ο αριθμός που υπάρχει μέσα στο όνομα αποτελεί αναγνωριστικό για το μέγεθος του εικονιδίου. Το αρχείο png είναι μια εικόνα η οποία δημιουργήθηκε για την εφαρμογή με χρήση του ελεύθερου προγράμματος διανυσματικών γραφικών Inkscape. Το εικονίδιο στη μορφή xpm δημιουργήθηκε από την παραπάνω εικόνα με χρήση του ελεύθερου προγράμματος imagemagick. Ένα αρχείο xpm είναι αρχείο εικόνας το οποίο περιέχει μόνο κώδικα στη γλώσσα προγραμματισμού C, ορίζοντας έναν πίνακα που περιέχει σε κάθε θέση του το χρώμα από το εκάστοτε εικονοστοιχείο (pixel) της εικόνας. Τέλος, στο φάκελο po βρίσκονται τα αρχεία με τις μεταφράσεις της εφαρμογής και στο φάκελο src ο πηγαίος της κώδικας.

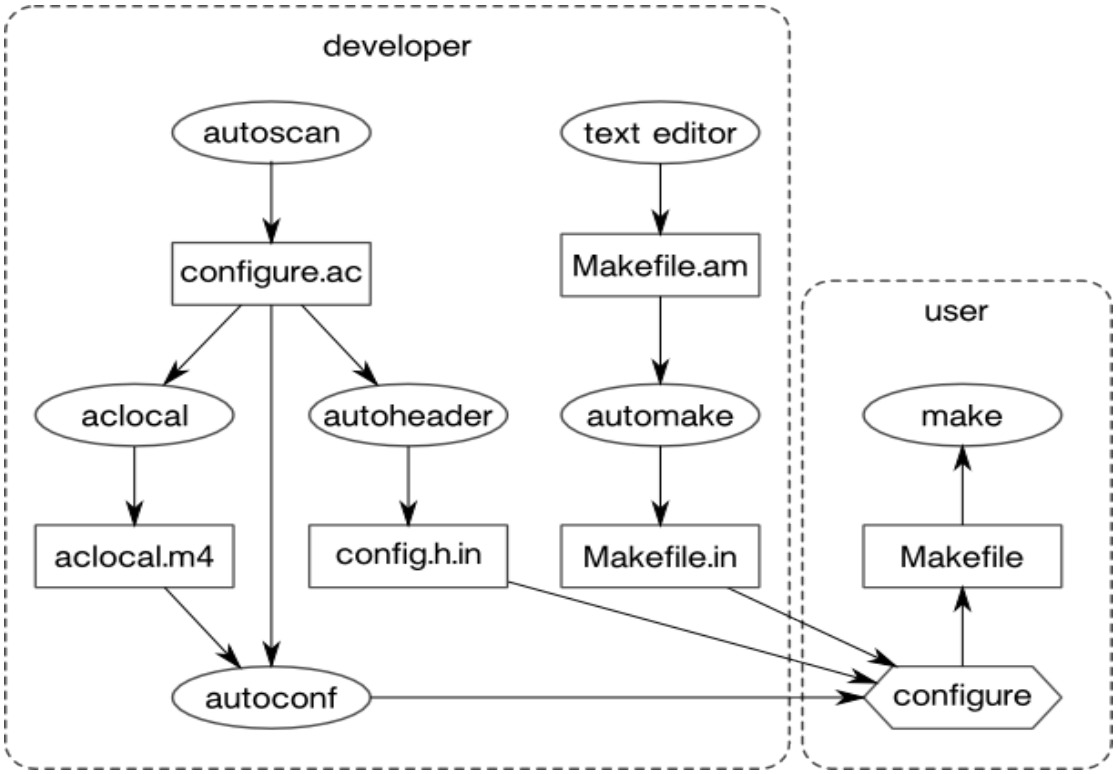
Όλα τα παραπάνω αρχεία των φακέλων πρέπει να τοποθετηθούν στους σωστούς καταλόγους του συστήματος. Στο κάθε Makefile.am μπορούμε εύκολα να το καθορίσουμε αυτό χρησιμοποιώντας της μεταβλητές που μας παρέχει το GNU Build System. Για παράδειγμα, κατά τη διαδικασία εκτέλεσης του configure script που παράγεται από το autoconf, ο χρήστης μπορεί να ορίσει τη μεταβλητή datadir κάνοντας χρήση του ορίσματος --prefix. Στις περισσότερες διανομές Linux αυτή η μεταβλητή από προεπιλογής θα δείχνει στον κατάλογο /usr. Έτσι, χρησιμοποιώντας τη μεταβλητή datadir μπορούμε εύκολα να τοποθετήσουμε τη συντόμευση της εφαρμογής στον κατάλογο $(datadir)/applications και τα εικονίδια στο $(datadir)/pixmaps. Για το man page υπάρχει η μεταβλητή mandir, την οποία αρκεί να συμπληρώσουμε με το κατάλληλο όνομα του φακέλου, ανάλογα τον τύπο της εφαρμογής. Εφόσον η εφαρμογή Giortes είναι ένα απλό πρόγραμμα και όχι κάποια βιβλιοθήκη ή κλήση συστήματος κλπ, το man page θα πρέπει να τοποθετηθεί στον κατάλογο $(mandir)/man1.

Στον φάκελο src, το Makefile.am είναι διαφορετικό από τα προηγούμενα καθώς περιέχει οδηγίες για τη μεταγλώττιση του πακέτου. Τέτοιες οδηγίες είναι η λίστα με τα αρχεία του πηγαίου κώδικα και οι ορισμοί που πρέπει να περαστούν κατά τη μεταγλώττιση και το link. Για παράδειγμα, το πρόγραμμα χρησιμοποιεί κάποιο εικονίδιο το οποίο πρέπει να το βρει από τον κατάλογο των εικονιδίων του συστήματος. Επειδή όμως αυτός ο κατάλογος δεν είναι ίδιος σε κάθε σύστημα, κατά τη μεταγλώττιση τον ορίζουμε ώστε να τον γνωρίζει έπειτα και η εφαρμογή. Επιπλέον οδηγίες που υπάρχουν σε αυτό το Makefile.am είναι κάποιες επιλογές του μεταγλωττιστή, η -Wall και η -O3. Η πρώτη κάνει τον μεταγλωττιστή να εμφανίζει όλες τις προειδοποιήσεις, ενώ η δεύτερη χρησιμοποιείται για να βελτιστοποιήσει το τελικό εκτελέσιμο. Τέλος, υπάρχει μια οδηγία προς τον linker, η -s, ώστε να παραλείπει όλα τα σύμβολα από το τελικό αρχείο. Το Makefile.am στη βάση του πακέτου περιέχει κάποιες γενικές οδηγίες για την εντολή automake καθώς και τους φακέλους στους οποίους θα βρει τα υπόλοιπα Makefiles.

Εφόσον έχουν δημιουργηθεί τα Makefile.am, με την εκτέλεση της εντολής automake η οποία τα δέχεται σαν είσοδο, παράγονται τα αρχεία Makefile.in. Από την πλευρά του προγραμματιστή το πρόγραμμα πλέον είναι έτοιμο για τη διανομή του. Εκτελώντας το configure script, μία από τις λειτουργίες του είναι να διαβάζει τα Makefile.in και να παράγει τα τελικά, βάσει του παρόντος συστήματος, αρχεία Makefile. Αυτά τα αρχεία περιέχουν πλέον τις σωστές τοποθεσίες του συστήματος και κάποιους κανόνες τους οποίους χρησιμοποιεί το πρόγραμμα make. Κάθε τέτοιος κανόνας είναι ένα σύνολο από λειτουργίες που πρέπει να εκτελεστούν. Οι πιο κοινοί κανόνες είναι ο install, ο οποίος εγκαθιστά το λογισμικό στο σύστημα, ο clean ο οποίος διαγράφει τα μη απαραίτητα αρχεία που δημιουργήθηκαν κατά τη μεταγλώττιση και ο uninstall ο οποίος

αφαιρεί το λογισμικό αν έχει ήδη εγκατασταθεί. Ένας ακόμη κανόνας, είναι ο dist, ο οποίος θα δημιουργήσει το τελικό πακέτο που θα περιέχει τον πηγαίο κώδικα και τα υπόλοιπα απαραίτητα αρχεία. Στο εξής, το πακέτο λογισμικού που δημιουργήθηκε με την εντολή make dist είναι έτοιμο και μπορεί να διανεμηθεί.

Πλέον ο τελικός χρήστης αρκεί να εκτελέσει το configure script ώστε να ελεγχθεί αν το σύστημά του είναι ικανό να μεταγλωττίσει το λογισμικό και να το προσαρμόσει κατάλληλα. Στη συνέχεια την εντολή make για τη μεταγλώττιση του προγράμματος και τέλος την εντολή make install για να εγκατασταθεί στο σύστημά του. Στην εικόνα 3.1 φαίνεται το διάγραμμα ροής των εργαλείων autoconf και automake από τη σκοπιά του προγραμματιστή και του τελικού χρήστη. Σε κύκλο παρουσιάζονται οι εντολές και σε ορθογώνιο τα αρχεία που χρησιμοποιούνται.



Εικόνα 3.2 : Το διάγραμμα ροής των εργαλείων autoconf και automake από τη σκοπιά του προγραμματιστή και του τελικού χρήστη.

3.4 Διεθνοποίηση του έργου με το GNU gettext

Λέγοντας διεθνοποίηση εννοούμε τη διαδικασία με την οποία ένα πακέτο λογισμικού, είναι σε θέση να υποστηρίζει πολλές γλώσσες. Πρόκειται για μια διαδικασία κατά την οποία τα προγράμματα αποδεσμεύονται από τη χρήση μόνο της αγγλικής γλώσσας και μπορούν να προσαρμοστούν ώστε να λειτουργήσουν με οποιαδήποτε. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν διάφορες τεχνικές για να διεθνοποιήσουν τα προγράμματά τους. Το GNU gettext είναι μια από αυτές. Είναι μια βιβλιοθήκη που βοηθά στη διεθνοποίηση δίνοντας τη δυνατότητα να μεταφραστεί η εφαρμογή σε πολλές γλώσσες.

Η βασική ιδέα της βιβλιοθήκης είναι να σημανθούν προς μετάφραση όλα τα αλφαριθμητικά που υπάρχουν στον πηγαίο κώδικα. Έπειτα, ένα προγραμματιστικό εργαλείο να εξάγει όλα αυτά τα αλφαριθμητικά σε ένα αρχείο, το οποίο να μεταφράζεται στις επιθυμητές γλώσσες. Στη συνέχεια, αυτό το αρχείο μεταγλωττίζεται σε αρχείο δυαδικής μορφής το οποίο εγκαθίσταται στο σύστημα μαζί με την εφαρμογή. Τέλος, κατά την εκτέλεση της εφαρμογής η βιβλιοθήκη libtool ψάχνει στο παραπάνω αρχείο για τα μεταφρασμένα αλφαριθμητικά της εφαρμογής. Εάν δεν υπάρχουν τότε θα εμφανιστούν όπως τα έγραψε ο προγραμματιστής μέσα στην εφαρμογή, διαφορετικά θα εμφανιστούν σύμφωνα με τη μετάφρασή τους στην εκάστοτε γλώσσα.

Πιο αναλυτικά, το πρώτο βήμα είναι η εκτέλεση της εντολής gettextize η οποία καθιστά το πακέτο λογισμικού ικανό να δέχεται μεταφράσεις και πληροφορεί για κάποιες επιπλέον τροποποιήσεις που πρέπει να γίνουν χειροκίνητα από τον προγραμματιστή. Αρχικά, πρέπει να προστεθεί στο αρχείο configure.ac η μακροεντολή AM_GNU_GETTEXT ώστε να ελεγχθεί αν το σύστημα έχει τη δυνατότητα να υποστηρίξει τις μεταφράσεις. Ακόμη, πρέπει να δημιουργηθεί στο φάκελο po το αρχείο Makevars από το πρότυπο αρχείο Makevars.template το οποίο περιέχει κάποιες βοηθητικές μεταβλητές και ορισμούς. Στο φάκελο με τον πηγαίο κώδικα πρέπει να προστεθεί στο αρχείο Makefile.am ο ορισμός της σταθεράς LOCALEDIR ώστε να γνωρίζει η εφαρμογή, με τη βοήθεια της βιβλιοθήκης libtool, που θα ψάξει για τα μεταφρασμένα αλφαριθμητικά.

Πλέον έχει ρυθμιστεί κατάλληλα το GNU Build System και μένει να ρυθμιστεί και ο πηγαίος κώδικας της εφαρμογής. Οπότε το επόμενο βήμα είναι η επισήμανση των αλφαριθμητικών προς μετάφραση στον πηγαίο κώδικα. Για να γίνει αυτό πρέπει τα αλφαριθμητικά να είναι γραμμένα στην Αγγλική γλώσσα ή σε οποιαδήποτε άλλη γλώσσα, αρκεί να χρησιμοποιούνται μόνο έγκυροι ascii χαρακτήρες. Αυτά τα αλφαριθμητικά επισημαίνονται με χρήση μιας ειδικής συνάρτησης με όνομα gettext. Όπως μας πληροφόρησε νωρίτερα και η εντολή gettextize,

χρειάζεται να προστεθεί στον πηγαίο κώδικα ένα επιπλέον header file το οποίο θα περιέχει τη δήλωση αυτής της συνάρτησης. Αυτό το header file, το gettext.h, υπάρχει ήδη στο σύστημα και μπορεί απλά να αντιγραφεί στο φάκελο του πηγαίου κώδικα. Χάρη ευκολίας είναι κοινή τακτική ο προγραμματιστής να ορίζει με ένα διαφορετικό και πιο σύντομο όνομα την παραπάνω συνάρτηση, συνήθως "_" με χρήση της #define _(x) gettext(x). Για να αρχικοποιηθεί κατάλληλα η βιβλιοθήκη gettext κατά την εκτέλεση της εφαρμογής, πρέπει να προστεθούν στον πηγαίο κώδικα οι συναρτήσεις setlocale, bindtextdomain και textdomain. Πλέον, μένει να επισημανθούν τα αλφαριθμητικά προς μετάφραση. Κάτι τέτοιο γίνεται εύκολα αλλάζοντας για παράδειγμα κάποια συνάρτηση printf("Hello World") σε printf(_("Hello world")).

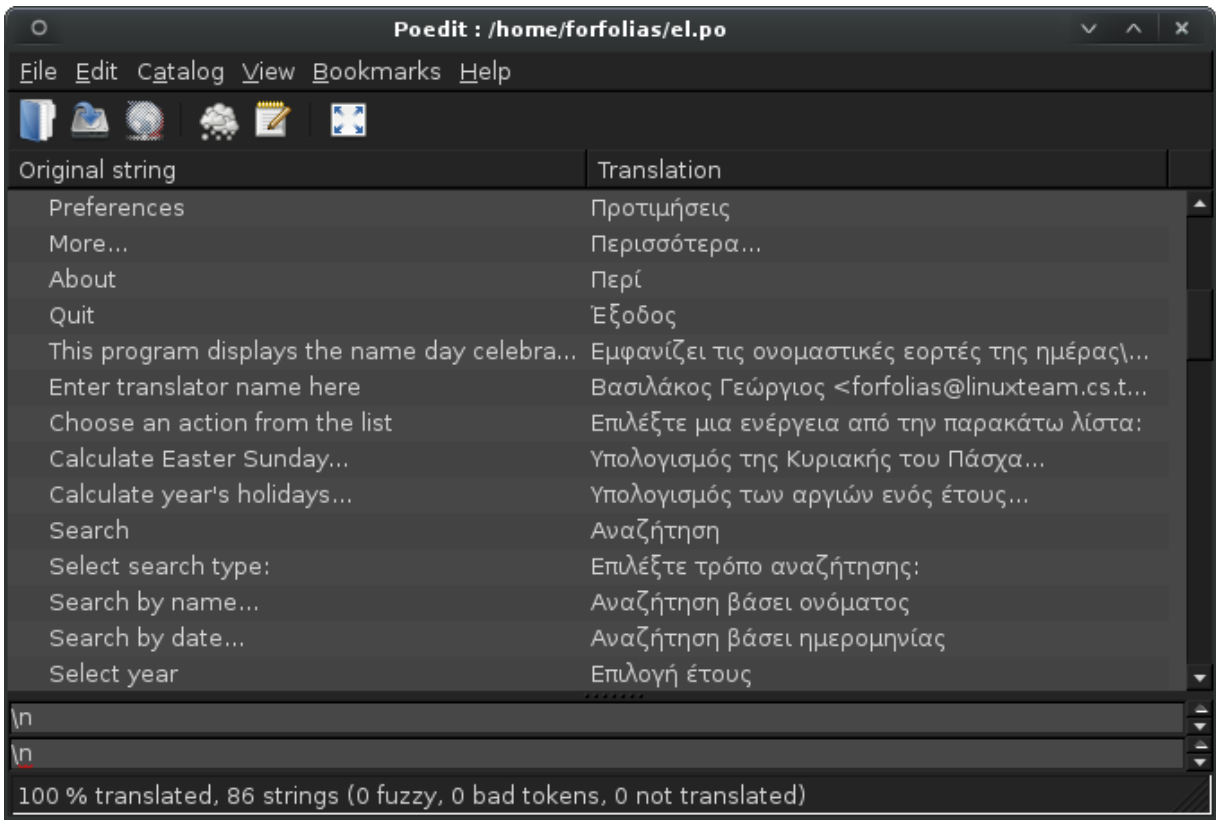
Όταν εκτελείται η εντολή make update-po γίνεται μια σειρά από κάποιες πολύ χρήσιμες ενέργειες. Μια από αυτές είναι η συλλογή των ονομάτων των αρχείων του πηγαίου κώδικα που περιέχουν αλφαριθμητικά προς μετάφραση και η εγγραφή τους στο αρχείο POTFILES που βρίσκεται στο φάκελο po. Ακόμη, η παραπάνω εντολή δημιουργεί ένα αρχείο .pot (portable object template), στην περίπτωση του προγράμματος Giortes έχει το όνομα giortes.pot, το οποίο περιέχει όλα τα αλφαριθμητικά προς μετάφραση και θα είναι το αρχείο που θα έχουν ως πρότυπο οι μεταφραστές.

Για να ξεκινήσει η μετάφραση πρέπει πρώτα να οριστεί στο αρχείο LINGUAS του φακέλου po το αναγνωριστικό της κάθε γλώσσας στην οποία θα μεταφραστεί η εφαρμογή. Στη συνέχεια πρέπει να δημιουργηθεί από το πρότυπο αρχείο .pot, το αρχείο με όνομα το αναγνωριστικό της γλώσσας και επέκταση .po (portable object), το οποίο θα περιέχει τη μετάφραση για τη γλώσσα αυτή. Για την ελληνική γλώσσα το αναγνωριστικό αυτό είναι το el οπότε το αρχείο έχει όνομα el.po. Αν γίνουν αλλαγές στον πηγαίο κώδικα της εφαρμογής, με αποτέλεσμα να αλλάξουν ή προστεθούν κάποια αλφαριθμητικά, τότε εκτελώντας ξανά την εντολή make update-po, η οποία εσωτερικά καλεί την εντολή msgmerge, ανανεώνεται το πρότυπο αρχείο .pot καθώς και όλα τα αρχεία μεταφράσεων.

Στις πρώτες γραμμές του αρχείου .pot υπάρχουν πληροφορίες όπως το όνομα του πακέτου λογισμικού προς μετάφραση, η ημερομηνία τελευταίας επεξεργασίας του αρχείου, η ομάδα μετάφρασης και κάποιο email επικοινωνίας με αυτή. Μετά από τις παραπάνω πληροφορίες, ακολουθούν τα αλφαριθμητικά προς μετάφραση με μια συγκεκριμένη δομή. Αυτή η δομή είναι αρχικά να υπάρχει η τοποθεσία του αλφαριθμητικού μέσα στον πηγαίο κώδικα, έπειτα κάποια σήμανση για τη γλώσσα του πηγαίου κώδικα, κάποιο αναγνωριστικό για το αλφαριθμητικό και τέλος η μετάφρασή του. Το αναγνωριστικό γίνεται χρησιμοποιώντας τη λέξη msgid ακολουθούμενη

από το αλφαριθμητικό όπως είναι στον πηγαίο κώδικα της εφαρμογής, ενώ για το μεταφρασμένο αλφαριθμητικό χρησιμοποιείται η λέξη msgstr ακολουθούμενη από τη μετάφρασή του.

Τα αρχεία .po είναι απλά αρχεία κειμένου τα οποία μπορούν να επεξεργαστούν με οποιονδήποτε τρόπο. Ωστόσο, έχουν αναπτυχθεί αρκετές εφαρμογές ελεύθερου λογισμικού για αυτό το σκοπό. Μια από αυτές τις εφαρμογές, η οποία χρησιμοποιήθηκε για τη μετάφραση του προγράμματος Giortes, είναι το Poedit, όπως φαίνεται στην εικόνα 3.3. Σε αυτό το σημείο η μετάφραση του λογισμικού μπορεί να γίνει ακόμη και από άτομα τα οποία δεν έχουν γνώσεις προγραμματισμού.

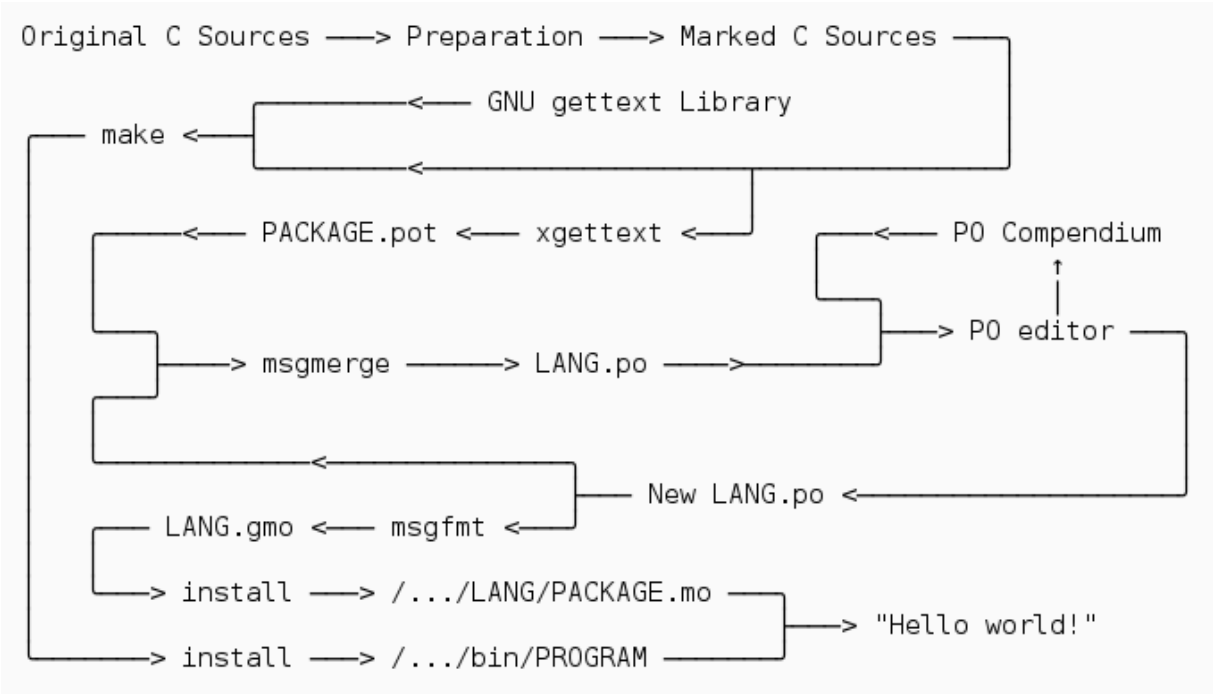


Εικόνα 3.3: Πρόγραμμα επεξεργασίας αρχείων po

Εφόσον έχει ολοκληρωθεί η μετάφραση του αρχείου .po, πρέπει στη συνέχεια να μεταγλωττιστεί σε αρχείο δυαδικής μορφής .mo (machine object). Κάτι τέτοιο γίνεται πάλι με την εκτέλεση της εντολής make update-po, η οποία εσωτερικά καλεί την εντολή msgfmt.

Το τελευταίο στάδιο για να λειτουργήσει η διεθνοποίηση του λογισμικού, είναι η τοποθέτηση του αρχείου .mo στο σωστό κατάλογο του συστήματος ώστε να μπορεί να βρεθεί από

τη βιβλιοθήκη gettext κατά την εκτέλεση. Στις περισσότερες διανομές Linux, αυτός ο κατάλογος είναι ο "/usr/share/locale/LL/LC_MESSAGES/", όπου LL είναι το αναγνωριστικό της γλώσσας. Για παράδειγμα, στην εφαρμογή Giortes, για την Ελληνική γλώσσα το αρχείο .mo πρέπει να τοποθετηθεί στον κατάλογο "/usr/share/locale/el/LC_MESSAGES/giortes.mo".



Εικόνα 3.4: Διάγραμμα ροής του εργαλείου gettext

Λόγω της φύσης του ελεύθερου λογισμικού, τις περισσότερες φορές ο μεταφραστής δεν έχει άμεση σχέση με τα υπόλοιπα στάδια ανάπτυξης του λογισμικού. Έτσι, ο προγραμματιστής επικεντρώνεται στη συγγραφή κώδικα, ο μεταφραστής στη διεθνοποίησή του και το λογισμικό εξελίσσεται ταχύτατα. Όπως ο προγραμματιστής δε μπορεί να ελέγξει την ορθότητα όλων των μεταφράσεων, συνήθως και μεταφραστής δεν έχει γνώσεις προγραμματισμού ώστε να ελέγχει τι ακριβώς μεταφράζει. Για αυτό το λόγο, ο προγραμματιστής πρέπει να διασφαλίζει την ομαλή συνεργασία τους χρησιμοποιώντας σωστές τεχνικές ανάπτυξης κώδικα και αποφεύγοντας λάθη που θα κάνουν δύσκολη τη ζωή του μεταφραστή.

Ένα από τα πιο συχνά λάθη το οποίο πρέπει να αποφεύγεται είναι διαχωρισμός μιας πρότασης σε δύο ή περισσότερα μέρη. Ενώ είναι αρκετά εύκολο για τον προγραμματιστή να κατανοήσει ολόκληρη την πρόταση, ο μεταφραστής θα δυσκολευτεί να μεταφράσει έτσι ώστε να ενώνονται κατάλληλα τα μέρη της. Ακόμη, μπορεί κάποιο κομμάτι της πρότασης να

χρησιμοποιείται και σε άλλο σημείο της εφαρμογής αλλά με διαφορετική έννοια, και το τελικό αποτέλεσμα να είναι εντελώς διαφορετικό από αυτό που αναμενόταν.

Ένα ακόμη συχνό πρόβλημα είναι όταν ο προγραμματιστής έχει στην ίδια πρόταση δύο ή περισσότερα σημεία εισαγωγής μεταβλητών. Αν για παράδειγμα υπάρχει ο ακόλουθος κώδικας printf("Press OK to %s %d file(s)", action, number), με τη μεταβλητή action να παίρνει τις τιμές create και delete, σε μερικές γλώσσες ίσως χρειαστεί να αλλάξουν σειρά τα σημεία εισαγωγής των μεταβλητών για να αποδοθεί σωστά η μετάφραση. Ο κώδικας στη Γερμανική γλώσσα θα ήταν printf("Drücken Sie OK, um %d Datei(en) zu %s", number, action). Ωστόσο επειδή ο μεταφραστής μπορεί να επέμβει μόνο στο πρώτο μέρος της συνάρτησης printf, αν αλλάξει τη σειρά από τα σημεία εισαγωγής αλλά όχι και τη σειρά των μεταβλητών, θα δημιουργηθούν σφάλματα κατά τη μεταγλώττιση. Η λύση σε ένα τέτοιο πρόβλημα είναι η διάσπαση σε δύο απλούστερες προτάσεις. Δηλαδή, μια printf(_(“Press OK to create %d file(s)”), number) για τη δημιουργία και άλλη μια για τη διαγραφή printf(_(“Press OK to delete %d file(s)”), number).

Ένα λάθος που επίσης πρέπει να αποφεύγεται είναι η χρήση λέξεων με διπλή έννοια. Για παράδειγμα, αν σε μια εφαρμογή υπάρχουν οι λέξεις προς μετάφραση “left”, “right”, “wrong” και “right”, η λέξη right θα περιέχεται μόνο μια φορά στο αρχείο .pot, οπότε θα μεταφραστεί με μία από τις δυο έννοιες που έχει. Συνήθως ο μεταφραστής είναι ο πρώτος που αντιλαμβάνεται τέτοιου είδους λάθη. Η ευκολότερη λύση είναι η χρήση μοναδικών λέξεων ή κάποια περίφραση. Μια άλλη λύση είναι η χρήση προθέματος και πριν την εμφάνιση της λέξης να απορρίπτεται, με χρήση κάποιων συναρτήσεων, ό,τι υπάρχει πριν το πρόθεμα. Για παράδειγμα, printf(_(“direction|right”))

Αναφορές

http://www.gnu.org/software/autoconf/manual/autoconf.html

http://www.gnu.org/software/make/manual/make.html

http://www.gnu.org/software/gettext/manual/gettext.html

http://www.freesoftwaremagazine.com/books/agaal/brief_introduction_to_gnu_autotools

http://www.linuxjournal.com/article/3023?page=0,1

http://sources.redhat.com/autobook/

http://sources.redhat.com/automake/automake.html

http://en.wikipedia.org/wiki/GNU_build_system

http://en.wikipedia.org/wiki/Autoconf

http://en.wikipedia.org/wiki/Automake

http://en.wikipedia.org/wiki/Makefile

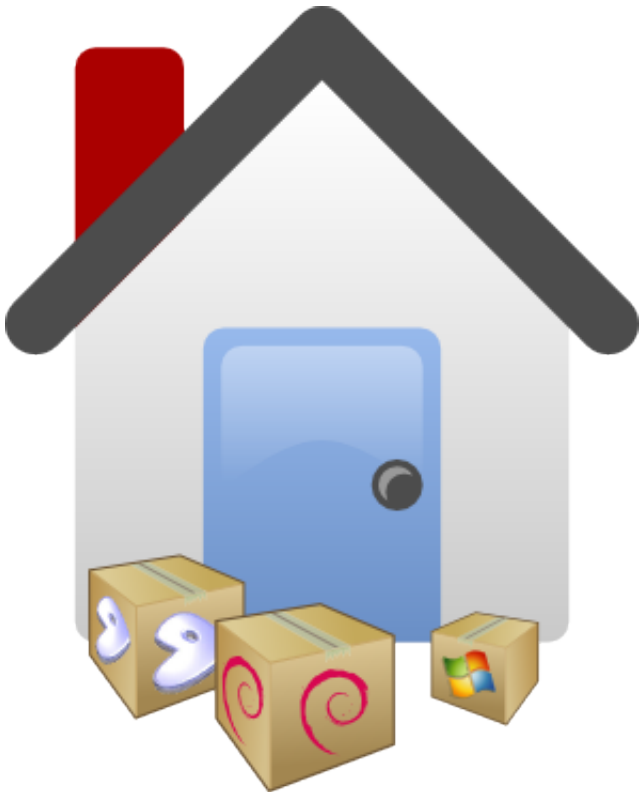
http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-1.0.html

http://en.wikipedia.org/wiki/X_PixMap

Managing Projects with GNU Make, Nutshell Handbooks, O'Reilly Media; 3 edition (November 19, 2004), ISBN-13: 978-0596006105

ΚΕΦΑΛΑΙΟ 4º

Διανομή του έργου



Μια διανομή ενός λογισμικού είναι ένα πακέτο από κάποιο συγκεκριμένο λογισμικό, ή μια συλλογή λογισμικού, ακόμα και ένα ολόκληρο λειτουργικό σύστημα, που έχει ήδη μεταγλωττιστεί και ρυθμιστεί. Είναι ένα πακέτο λογισμικού το οποίο συνήθως κατασκευάζεται από τους επίσημους δημιουργούς του λογισμικού και δίνεται στον τελικό χρήστη έτοιμο προς χρήση. Σε λειτουργικά συστήματα Microsoft Windows, συνήθως είναι ένα εκτελέσιμο πρόγραμμα εγκατάστασης (installer), ενώ στο λειτουργικό σύστημα Linux έχει τη μορφή πακέτου που περιέχει προρυθμισμένα τα απαραίτητα αρχεία.

4.1 Microsoft Windows

Τα Microsoft Windows είναι μια σειρά από λειτουργικά συστήματα για προσωπικούς υπολογιστές και διακομιστές. Είναι ένα περιβάλλον εργασίας για τους υπολογιστές, όπου η κύρια καινοτομία τους είναι ο εύκολος τρόπος επικοινωνίας του χρήστη με τον υπολογιστή.

Λογότυπο Microsoft Windows

Η Microsoft κυκλοφόρησε την πρώτη έκδοση των Windows το 1985 σαν ένα επιπρόσθετο πρόγραμμα για το λειτουργικό σύστημα MS-DOS με σκοπό τη δημιουργία γραφικής διεπαφής. Από το 1985 μέχρι σήμερα, η Microsoft έχει παρουσιάσει 13 εκδόσεις του λειτουργικού της συστήματος και έχει καταφέρει να επικρατήσει στην παγκόσμια αγορά προσωπικών υπολογιστών με ένα μερίδιο αγοράς που υπολογίζεται περίπου στο 90%. Τα Microsoft Windows είναι λογισμικό κλειστού κώδικα, ενώ η Microsoft έχει κατηγορηθεί αρκετές φορές για μονοπωλιακές τακτικές και επιθέσεις στο ελεύθερο λογισμικό. Ωστόσο, λόγω του πολύ μεγάλου μεριδίου αγοράς συνήθως τα περισσότερα προγράμματα στις μέρες μας έχουν ως προτεραιότητα τη συγκεκριμένη πλατφόρμα.

4.1.1. MinGW και MSYS

Το MinGW (Minimalist GNU for Windows) είναι μια μεταφορά του GNU Compiler Collection (GCC) στα λειτουργικά συστήματα Microsoft Windows, μαζί με ένα σύνολο από βιβλιοθήκες και header files. Το MinGW δίνει τη δυνατότητα στους προγραμματιστές να μεταφέρουν μια εφαρμογή γραμμένη για λειτουργικά συστήματα τύπου Unix σε Microsoft Windows. Το MSYS (Minimal SYStem) είναι μια συλλογή από εργαλεία του GNU όπως το bash, το make, το gawk και το grep τα οποία επιτρέπουν την ανάπτυξη εφαρμογών που εξαρτώνται από την παρουσία εργαλείων του UNIX. Σκοπός του MSYS είναι να συμπληρώσει τις ελλείψεις του

MinGW και της γραμμής εντολών των Windows. Ένα παράδειγμα χρήσης τους είναι η μεταγλώττιση μια εφαρμογής που στηρίζεται στο GNU Build System, εφόσον το configure script χρειάζεται το εργαλείο bash για να εκτελεστεί, το οποίο δεν υπάρχει στα Windows αλλά παρέχεται από το MSYS.

Τα εργαλεία που παρέχονται από το έργο MinGW μπορούν να χρησιμοποιηθούν είτε από τη γραμμή εντολών των Windows είτε να ενσωματωθεί σε κάποιο περιβάλλον ανάπτυξης λογισμικού (IDE). Το MinGW υποστηρίζει στατικές και δυναμικές βιβλιοθήκες σύμφωνα με την ονομασία <όνομα>.lib και <όνομα>.dll, σε σχέση με την κανονική ονοματολογία που ακολουθείται σε λειτουργικά συστήματα τύπου Unix, lib<όνομα>.a και lib<όνομα>.so. Η υλοποίηση των Win32 header files και Win32 στατικών βιβλιοθηκών κυκλοφορούν κάτω από ειδική άδεια χρήσης, ενώ οι μεταφορές των εργαλείων GNU είναι κάτω από την άδεια GNU General Public License.

Το έργο MinGW παρέχει τα παρακάτω πακέτα λογισμικού :

- binutils: Περιέχει τον assembler και τον linker, που μετατρέπουν τα παραγόμενα αρχεία του μεταγλωττιστή σε εκτελέσιμη (δυαδική) μορφή.
- mingw-runtime: Περιέχει τα header files και τις στατικές βιβλιοθήκες για τη βιβλιοθήκη της C, που απαιτούνται από εφαρμογές που έχουν μεταγλωττιστεί με το MinGW.
- w32api: Περιέχει τα Header files και τις στατικές βιβλιοθήκες για το λειτουργικό σύστημα Microsoft Windows. Παρέχει τη διασύνδεση (API) για την πρόσβαση στις βασικές λειτουργίες του λειτουργικού συστήματος.
- gcc: Είναι η μεταφορά του GNU Compiler Collection. Πρόκειται για το πρόγραμμα που αναλύει τον πηγαίο κώδικα και τον μεταγλωττίζει σε αντικείμενα (objects).
- mingw-gdb: Είναι η μεταφορά του GNU debugger, του προγράμματος που βοηθά στην αποσφαλμάτωση των εφαρμογών.
- mingw32-make: Είναι η μεταφορά του εργαλείου GNU make, του προγράμματος που αναλύει τα Makefiles.
- mingw-utils: Περιέχει διάφορα χρήσιμα προγραμματιστικά εργαλεία.

4.1.2. Nullsoft Scriptable Install System

Λογότυπο NSIS

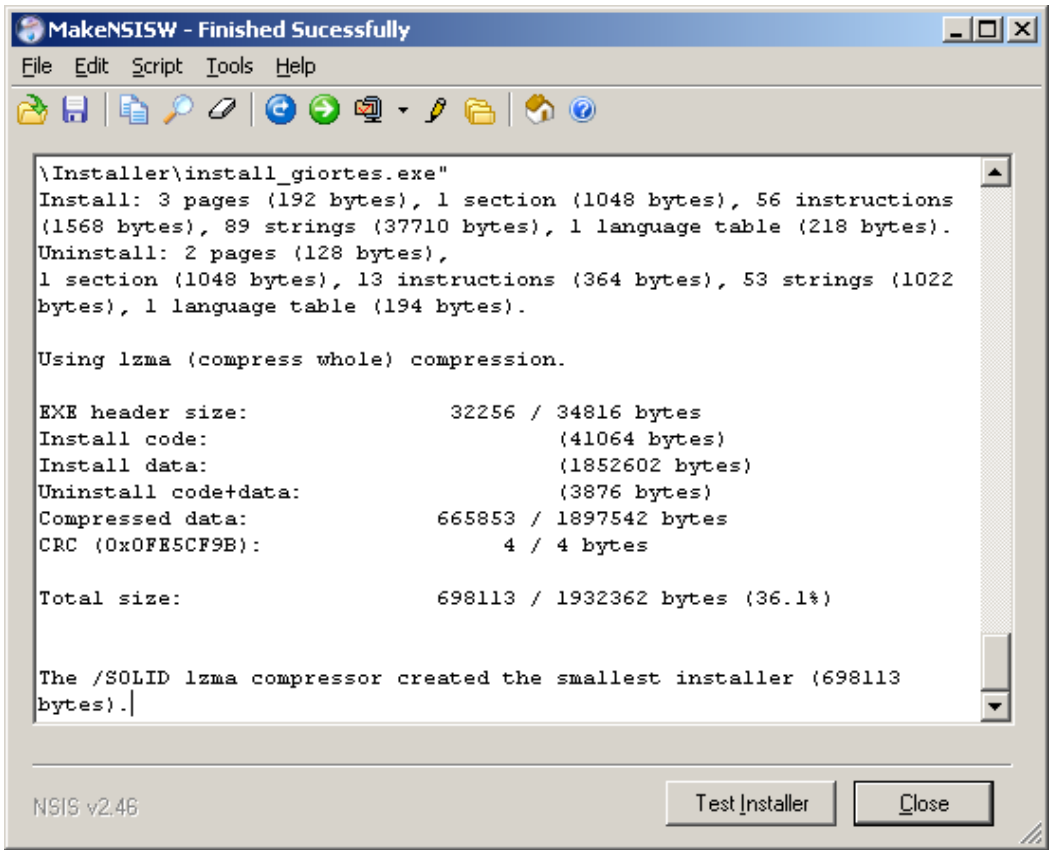
Τα περισσότερα πακέτα λογισμικού έρχονται με κάποιο πρόγραμμα που αναλαμβάνει τη σωστή εγκατάσταση του λογισμικού στο σύστημα, τον installer. Ένας installer σχεδόν αυτόματα αντιγράφει ή ανανεώνει αρχεία, γράφει κλειδιά στο μητρώο (registry) των Windows, δημιουργεί συντομεύσεις και εκτελεί άλλες παρόμοιες εργασίες. Το μόνο που χρειάζεται είναι κάποιες βασικές πληροφορίες από τον χρήστη και αυτός θα αναλάβει τα υπόλοιπα. Αυτό γίνεται συνήθως μέσω μιας εύχρηστης διεπαφής που δέχεται τις επιλογές του χρήστη και τον ενημερώνει για τις ενέργειες και την κατάσταση της εγκατάστασης. Μετά την εγκατάσταση της εφαρμογής, ο χρήστης το μόνο που μένει να κάνει είναι να εκτελέσει την εφαρμογή.

Το NSIS (Nullsoft Scriptable Install System) είναι ένα εργαλείο ελεύθερου λογισμικού το οποίο βοηθά τους προγραμματιστές στη δημιουργία installer για λειτουργικά συστήματα Windows. Δημιουργεί installers που είναι ικανοί να εγκαταστήσουν και απεγκαταστήσουν λογισμικό, να ρυθμίσουν κατάλληλα το σύστημα, να κατεβάσουν αρχεία από το διαδίκτυο, να αποσυμπιέσουν αρχεία και άλλα. Καθώς ο installer δημιουργείται βάσει αρχείων script (σενάρια) είναι πλήρως παραμετροποιήσιμος. Η γλώσσα των script είναι αρκετά εύκολη στην εκμάθηση και επιτρέπει τη χρήση μεταβλητών, συναρτήσεων, κάνει διαχείριση αλφαριθμητικών και άλλων ενεργειών όπως γίνεται και σε μια κανονική γλώσσα προγραμματισμού. Ο NSIS compiler μεταγλωττίζει τα αρχεία script και σε συνδυασμό με τα αρχεία του λογισμικού δημιουργεί ένα εκτελέσιμο πρόγραμμα εύκολο στη διανομή.

Η πρώτη ενέργεια για τη δημιουργία του installer είναι η συγγραφή των αρχείων script, τα οποία είναι απλά αρχεία κειμένου με κάποια ειδική σύνταξη. Μπορούν να δημιουργηθούν με οποιονδήποτε επεξεργαστή κειμένου και πρέπει να έχουν την επέκταση .nsi. Κάθε γραμμή του script αποτελεί μια εντολή και συνολικά το αρχείο περιέχει κάποια χαρακτηριστικά, τμήματα και συναρτήσεις. Τέτοια χαρακτηριστικά είναι η μεταβλητή outFile, η οποία πληροφορεί το NSIS για το όνομα που θα έχει ο installer, και η installDir για τον κατάλογο στον οποίο θα εγκατασταθεί η εφαρμογή. Ακόμη, υπάρχουν οι μεταβλητές LicenseData και LicenseText που αναλαμβάνουν την εμφάνιση της άδειας χρήσης του λογισμικού με την οποία πρέπει να συμφωνεί ο χρήστης για να συνεχίσει στην εγκατάστασή του. Κάθε installer πρέπει υποχρεωτικά να έχει τουλάχιστον ένα τμήμα (section) με κάποιες λειτουργίες. Για παράδειγμα, ένα τμήμα μπορεί να αφορά την εγκατάσταση του λογισμικού και ένα επιπλέον τμήμα την απεγκατάστασή του.

Οι εντολές, ή οδηγίες, που υπάρχουν μέσα στα τμήματα είναι διαφορετικές από τα χαρακτηριστικά καθώς ενεργοποιούνται κατά την εκτέλεση του installer. Αυτές οι οδηγίες είναι υπεύθυνες για την εγγραφή κλειδιών στο μητρώο των Windows, για τη δημιουργία αρχείων, καταλόγων, συντομεύσεων και άλλων λειτουργιών. Η πιο βασική οδηγία είναι η SetOutPath η οποία πληροφορεί τον installer για τον κατάλογο στον οποίο θα εξαχθούν τα αρχεία.

Οι συναρτήσεις επίσης περιέχουν εντολές σαν αυτές των τμημάτων. Η διαφορά τους είναι στον τρόπο με τον οποίο καλούνται. Υπάρχουν δυο ειδών συναρτήσεις, η user functions και οι callback fun functions. Οι user functions δε θα εκτελεστούν, εκτός αν κληθούν από τον χρήστη μέσα από κάποιο τμήμα ή άλλη συνάρτηση με χρήση της οδηγίας Call. Μετά την εκτέλεση των εντολών της συνάρτησης που κλήθηκε από την οδηγία Call, η εκτέλεση θα συνεχιστεί με την επόμενη εντολή μετά την Call εκτός και αν ακυρώθηκε η διαδικασία της εγκατάστασης μέσα στην παραπάνω συνάρτηση. Αυτού του είδους οι συναρτήσεις είναι χρήσιμες όταν ένα κομμάτι οδηγιών επαναλαμβάνεται συχνά σε διάφορα σημεία του installer. Οι callback functions είναι προαιρετικές και καλούνται από τον installer όταν γίνει κάποιο συγκεκριμένο γεγονός, όπως για παράδειγμα η εκκίνηση του installer.



Εικόνα 4.1: Το πρόγραμμα MakeNSIS που χρησιμοποιείται για τη μεταγλώττιση των αρχείων script και τη δημιουργία installer

Εφόσον έχει δημιουργηθεί το αρχείο script, το επόμενο βήμα είναι να μεταγλωττιστεί από την εφαρμογή MakeNSIS.exe, ώστε να δημιουργηθεί ο installer, όπως φαίνεται στην εικόνα 4.1. Ο μεταγλωττιστής θα διαβάσει το αρχείο script και αν υπάρχουν σφάλματα θα ειδοποιήσει τον προγραμματιστή και θα ακυρώσει τη μεταγλώττιση. Αν τα σφάλματα δεν είναι κρίσιμα η μεταγλώττιση θα συνεχιστεί παράγοντας το τελικό εκτελέσιμο αρχείο έτοιμο προς διανομή.

Για να εκτελεστεί σωστά η εφαρμογή Giortes σε περιβάλλον Microsoft Windows, πρέπει να είναι εγκατεστημένο το GTK+ Runtime Environment. Για αυτό το λόγο δημιουργήθηκαν δυο installers. Ο πρώτος installer ελέγχει αν υπάρχει το GTK+ Runtime Environment στο σύστημα και εάν δεν υπάρχει τον κατεβάζει από το διαδίκτυο και τον εγκαθιστά μαζί με την εφαρμογή. Ο δεύτερος έχει ενσωματωμένο το GTK+ Runtime Environment και το εγκαθιστά σαν κομμάτι του λογισμικού Giortes με τίμημα το λίγο μεγαλύτερο μέγεθος του τελικού installer.

4.2 Linux

Λογότυπο Linux

Η ονομασία Linux, που στα ελληνικά προφέρεται λίνουξ, είναι ένας γενικός όρος αναφοράς σε λειτουργικά συστήματα που βασίζονται στον πυρήνα Linux. Η αρχιτεκτονική του Linux είναι παρόμοια με αυτή του λειτουργικού συστήματος Unix αλλά έχει αναπτυχθεί εκ του μηδενός και δεν περιλαμβάνει κώδικα από το Unix. Η ανάπτυξη του Linux είναι χαρακτηριστικό παράδειγμα εθελοντικής συνεργασίας από διαδικτυακές κοινότητες. Όλο το έργο είναι ανοικτού κώδικα και ελεύθερα προσβάσιμο από όλους για αντιγραφή, τροποποίηση ή αναδιανομή χωρίς περιορισμό καθώς διανέμεται κάτω από την άδεια χρήσης GNU General Public License.

Το Linux μπορεί να εγκατασταθεί και να λειτουργήσει σε μεγάλη ποικιλία υπολογιστικών συστημάτων, από μικρές συσκευές όπως κινητά τηλέφωνα μέχρι μεγάλα υπολογιστικά συστήματα και υπερυπολογιστές. Κυκλοφορεί σε διανομές Linux, δηλαδή ένα σύνολο που περιέχει τον πυρήνα και συνοδευτικά προγράμματα, όπως βιβλιοθήκες, εργαλεία συστήματος, παραθυρικό περιβάλλον εργασίας και πολλές άλλες εφαρμογές που απαιτούνται για την εύρυθμη λειτουργία ενός υπολογιστή. Χαρακτηριστικό των διανομών είναι η μεγάλη δυνατότητα παραμετροποίησης και επιλογής που προσφέρουν καθώς υπάρχουν πολλές και η κάθε μια απευθύνεται σε διαφορετικό τύπο χρηστών. Ανάλογα με την φιλοσοφία που ακολουθεί κάθε διανομή μπορεί να δίνει μεγαλύτερη βάση στη φιλικότητα προς τον χρήστη, στις εφαρμογές πολυμέσων, την ευκολία παραμετροποίησης και αλλού.

Δημιουργός του Linux είναι ο Linus Torvalds, από το όνομα του οποίου προήλθε και η ονομασία Linux. Ο Torvalds άρχισε να αναπτύσσει τον πυρήνα το 1991 εμπνευσμένος από το λειτουργικό MINIX και χρησιμοποιώντας ελεύθερες βιβλιοθήκες από το GNU Project που είχε δημιουργήσει νωρίτερα ο Richard Stallman. Λόγω των στενών σχέσεων μεταξύ Linux και GNU, το λειτουργικό σύστημα πολλές φορές αναφέρεται ως GNU/Linux, ονομασία που προτιμά και το Ίδρυμα Ελεύθερου Λογισμικού.

4.2.1. Η διανομή Debian GNU/Linux

Το Debian, αποτέλεσμα του Debian Project, είναι μια δημοφιλής διανομή Linux, ελεύθερο λογισμικό που αναπτύσσεται μέσω της συνεργασίας εθελοντών από όλο τον κόσμο. Ξεκίνησε να αναπτύσσεται το 1993 από τον Ian Murdock και βασίζεται στον πυρήνα linux και στην ομάδα βασικών εργαλείων του εγχειρήματος GNU.

Λογότυπο διανομής Debian

Το Debian είναι γνωστό για την αφοσίωσή του στη φιλοσοφία του Unix και του ελεύθερου λογισμικού. Είναι επίσης γνωστό για το πλήθος επιλογών και δυνατοτήτων που προσφέρει. Η τρέχουσα έκδοση περιλαμβάνει πάνω από 25.000 πακέτα λογισμικού για δεκαπέντε αρχιτεκτονικές υπολογιστών που το φάσμα τους κυμαίνεται από ARM αρχιτεκτονική, που διαθέτουν συνήθως ενσωματωμένα συστήματα μέχρι τις πιο κοινές x86 και PowerPC αρχιτεκτονικές που υπάρχουν στους μοντέρνους προσωπικούς υπολογιστές.

Το Debian είναι επίσης πολύ γνωστό για το σύστημα διαχείρισης πακέτων που διαθέτει, το APT Advanced Packaging Tool και τις αυστηρές πολιτικές που υιοθετεί ως προς την ποιότητα των πακέτων και των εκδόσεων του και την ανοιχτή διαδικασία ανάπτυξης και ελέγχου που υιοθετεί. Αυτές οι πρακτικές κάνουν πιο εύκολες τις αναβαθμίσεις και την εγκατάσταση ή αφαίρεση πακέτων. Το Debian υποστηρίζεται από δωρεές που γίνονται μέσω οργανισμών που προωθούν το ελεύθερο λογισμικό. Το Debian δεν υποστηρίζεται από κάποια εταιρία, αλλά από το Debian Project και τον οργανισμό Software in the Public Interest.

4.2.1.1 Εκδόσεις και αποθετήρια

Το έργο Debian προσφέρει ταυτόχρονα τρεις εκδόσεις της διανομής, καθεμιά με διαφορετικά χαρακτηριστικά και για διαφορετικούς σκοπούς. Αυτές οι εκδόσεις περιλαμβάνουν στα κύρια αποθετήρια λογισμικού (repositories) πακέτα τα οποία συμμορφώνονται με το Debian Free Software Guidelines (DFSG). Το DFSG είναι μια σειρά από κατευθυντήριες γραμμές που χρησιμοποιεί το έργο Debian για να καθορίσει αν μια άδεια χρήσης λογισμικού είναι άδεια χρήσης ελεύθερου λογισμικού, το οποίο είναι καθοριστικός παράγοντας για το αν το λογισμικό μπορεί να περιληφθεί στο Debian.

Η έκδοση stable είναι πάντα η παρούσα επίσημη κυκλοφορία της διανομής και περιέχει πολύ σταθερό και δοκιμασμένο λογισμικό. Η έκδοση stable δημιουργείται παγώνοντας την έκδοση testing για κάποιους μήνες και διορθώνοντας συνεχώς τα σφάλματά της ώστε να γίνει όσο πιο σταθερή γίνεται. Όταν φτάσει να μην έχει πολλά και σημαντικά σφάλματα τότε κυκλοφορεί σαν έκδοση stable. Οι ενημερώσεις σε αυτή την έκδοση αφορούν μόνο σημαντικά κενά ασφαλείας και χρηστικότητας. Αυτή η έκδοση συνήθως χρησιμοποιείται σε διακομιστές (servers).

Η έκδοση testing είναι πάντα η επόμενη κύρια κυκλοφορία της διανομής και όπως μαρτυρά και το όνομά της είναι κάτω από συνεχή έλεγχο. Τα πακέτα αυτής της έκδοσης έχουν υποστεί κάποιους ελέγχους, καθώς προέρχονται από την έκδοση unstable, αλλά δεν πληρούν τις προϋποθέσεις ώστε να είναι στη σταθερή έκδοση. Τα πακέτα αυτής της έκδοσης είναι πιο νέα από της σταθερής αλλά πιο παλιά από την έκδοση unstable. Αυτή η έκδοση ανανεώνεται συνεχώς, έως ότου μπει σε κατάσταση παγώματος και αρχίσει η μεθοδική αποσφαλμάτωσή της, και συνίσταται για οικιακή χρήση.

Η έκδοση unstable παρέχει πακέτα λογισμικού τα οποία είναι υπό εξέλιξη και ανανεώνονται συνεχώς. Αυτή η έκδοση είναι σχεδιασμένη για προγραμματιστές που συμμετέχουν σε έργα και χρειάζονται τις πιο πρόσφατες εκδόσεις βιβλιοθηκών και προγραμμάτων. Αυτή η έκδοση αλλάζει συνεχώς και δεν είναι τόσο σταθερή όσο οι υπόλοιπες, για αυτό δεν κυκλοφορούν επίσημα CD και DVD.

Τα Debian Free Software Guidelines (DFSG) επιμένουν σε μια σχετικά αυστηρή ερμηνεία του ελεύθερου λογισμικού, ωστόσο δεν έχουν εγκριθεί ακόμα από το Ίδρυμα Ελεύθερου Λογισμικού καθώς υποστηρίζουν και περιλαμβάνουν ένα αποθετήριο με κλειστού κώδικα λογισμικό. Αποθετήρια λογισμικού είναι ένας αποθηκευτικός χώρος με πακέτα λογισμικού τα οποία μπορούν να ανακτηθούν και να εγκατασταθούν σε έναν υπολογιστή μέσω του διαχειριστή πακέτων

της κάθε διανομής. Σύμφωνα με τις παραπάνω κατευθυντήριες γραμμές, ένας μικρός αριθμός πακέτων λογισμικού αποκλείονται από τα κύρια αποθετήρια της διανομής και περιλαμβάνονται στα αποθετήρια non-free και contrib, τα οποία δεν είναι επίσημο κομμάτι του έργου Debian.

Το αποθετήριο contrib περιλαμβάνει πακέτα λογισμικού που δεν πληρούν τις προϋποθέσεις του DFSG. Ο συνηθέστερος λόγος είναι όταν ένα πακέτο ελεύθερου λογισμικού χρησιμοποιεί ή χρειάζεται ένα πακέτο λογισμικού κλειστού κώδικα. Το αποθετήριο non-free περιλαμβάνει πακέτα τα οποία είναι κλειστού κώδικα λογισμικό αλλά είναι ελεύθερη η διανομή τους. Ακόμη, υπάρχει το αποθετήριο experimental, το οποίο ουσιαστικά δεν είναι ένα ολοκληρωμένο αποθετήριο αλλά προσωρινό για κάποιο πειραματικό λογισμικό. Τυχόν εξαρτήσεις του experimental αποθετηρίου συνήθως καλύπτονται από το unstable, ενώ το έργο Debian προειδοποιεί ότι τα πακέτα σε αυτό είναι ασταθή και με πολλά σφάλματα. Τέλος, υπάρχει το αποθετήριο volatile που παρέχει ενημερώσεις στην έκδοση stable για προγράμματα τα οποία χρειάζονται συνεχείς ενημερώσεις.

4.2.1.2 Debian Policy και Debian build toolchain

Το Debian Policy είναι ένα κείμενο που περιγράφει τις απαιτήσεις που έχει η πολιτική που ακολουθεί το έργο Debian. Αυτές οι απαιτήσεις αναφέρονται στη δομή και τα περιεχόμενα της διανομής και σε διάφορα άλλα θέματα σχεδιασμού του λειτουργικού συστήματος. Ακόμη, περιγράφει τις τεχνικές προδιαγραφές που πρέπει να πληρεί ένα πακέτο λογισμικού ώστε να συμπεριληφθεί στη διανομή.

Το Debian build toolchain είναι μια συλλογή από προγραμματιστικά εργαλεία που χρησιμοποιούνται για τη δημιουργία πακέτων πηγαίου κώδικα για το Debian, με κατάληξη .dsc, και πακέτων που περιέχουν προγράμματα εκτελέσιμης μορφής, με κατάληξη .deb, από επίσημα πακέτα πηγαίου κώδικα μια εφαρμογής (tarball). Αυτά τα εργαλεία χρησιμοποιούνται τόσο στο έργο Debian όσο και σε άλλες διανομές που βασίζονται σε αυτό, όπως το Ubuntu.

Ο πηγαίος κώδικας για εφαρμογές ελεύθερου λογισμικού συνήθως διανέμεται σε συμπιεσμένα αρχεία tar, τα οποία ονομάζονται tarball. Το Debian είναι μια διανομή προσανατολισμένη στην εκτελέσιμη μορφή των προγραμμάτων, πράγμα που σημαίνει ότι τα πακέτα .deb θα περιλαμβάνουν ήδη μεταγλωττισμένα προγράμματα και τα δεδομένα θα είναι οργανωμένα σε μια ιεραρχία όπως την περιμένει η εφαρμογή. Για αυτό το λόγο το Debian build toolchain χρειάζεται κάποιες οδηγίες για το πώς θα δημιουργήσει το πακέτο .deb από το επίσημο πακέτο του πηγαίου κώδικα.

Αυτές οι οδηγίες αποθηκεύονται σε έναν υποκατάλογο με όνομα debian, ο οποίος βρίσκεται στο δέντρο του πηγαίου κώδικα του πακέτου, από τον προγραμματιστή που δημιουργεί το πακέτο .deb. Αν και είναι δυνατό να κατασκευαστεί απευθείας το πακέτο από το διαμορφωμένο κατάλογο του αρχικού πηγαίου κώδικα, είναι κοινή πρακτική να δημιουργείται ένα επιπλέον πακέτο με τον πηγαίο κώδικα που θα περιέχει και τις αλλαγές που έκανε ο κατασκευαστής του πακέτου.

Ένα τυπικό πακέτο Debian πηγαίου κώδικα αποτελείται από τρία αρχεία :

- Το επίσημο πακέτο με τον πηγαίο κώδικα. Αυτό μπορεί να είναι είτε ένα αντίγραφο του επίσημου πακέτου με τον πηγαίο κώδικα αν δεν χρειάζονται αλλαγές, είτε ένα νέο πακέτο αν χρειάζονται αλλαγές ώστε να συμμορφώνεται με το Debian Free Software Guidelines.

- Ένα αρχείο diff.gz, το οποίο περιέχει τις αλλαγές που έγιναν από το δημιουργό του πακέτου στο επίσημο πακέτο. Πιο αναλυτικά, περιλαμβάνει όλο τον υποκατάλογο debian/ και τυχόν αλλαγές εκτός αυτού και είναι σε μορφή αρχείου diff συμπιεσμένο με το εργαλείο gzip.

- Το αρχείο .dsc, το οποίο είναι ένα αρχείο κειμένου με κάποια μεταδεδομένα όπως για παράδειγμα τα ονόματα όλων των αρχείων που αποτελούν το πακέτο του πηγαίου κώδικα και τα MD5 checksums τους. Ακόμη, περιέχει την ψηφιακή υπογραφή του δημιουργού του πακέτου του πηγαίου κώδικα.

Για παράδειγμα, στην εφαρμογή Giortes στην έκδοση 1.3, τα αρχεία θα είναι τα giortes_1.3.orig.tar.gz, giortes_1.3-1.diff.gz και giortes_1.3-1.dsc.

Το πακέτο με τον πηγαίο κώδικα δημιουργείται με το εργαλείο dpkg-buildpackage. Αυτό εκτελεί μια σειρά από ενέργειες όπως το να καλεί το αρχείο rules, καθαρίζει το πακέτο από τα περιττά αρχεία, κάνει διάφορους ελέγχους και υπογράφει το τελικό πακέτο με την υπογραφή του δημιουργού του πακέτου.

Στον κατάλογο debian περιέχονται αρχεία τα οποία χρησιμοποιούνται από το εργαλείο dpkg-buildpackage για τη δημιουργία τόσο του πακέτου με τον πηγαίο κώδικα όσο για αυτό που θα περιέχει το πρόγραμμα στην εκτελέσιμή του μορφή. Τουλάχιστον τρία αρχεία σε αυτό τον κατάλογο είναι απαραίτητα για τη σωστή δημιουργία του πακέτου, το αρχείο changelog, το control και το rules, ωστόσο για την εφαρμογή Giortes χρησιμοποιήθηκαν περισσότερα. Όλα τα παραπάνω αρχεία είναι απλά αρχεία κειμένου και μπορούν να δημιουργηθούν από οποιονδήποτε επεξεργαστή

κειμένου. Πρέπει να έχουν μια συγκεκριμένη δομή η οποία αναγράφεται αναλυτικά στο κείμενο Debian Policy.

Το αρχείο debian/changelog περιέχει πληροφορίες για όλες τις εκδόσεις του πακέτου από τότε που δημιουργήθηκε. Τα εργαλεία για την κατασκευή του πακέτου επεξεργάζονται μόνο την κορυφαία εισαγωγή, η οποία καθορίζει την έκδοση του πακέτου, την αναγκαιότητά του και τα σφάλματα που διορθώνει η συγκεκριμένη έκδοση του πακέτου στην παρούσα διανομή.

Το αρχείο debian/control περιέχει πληροφορίες για το πακέτο του πηγαίου κώδικα και τα πακέτα σε εκτελέσιμη μορφή που θα δημιουργηθούν από αυτό. Πιο αναλυτικά περιέχει πληροφορίες όπως το όνομα του πακέτου, το δημιουργό του, την αρχιτεκτονική επεξεργαστή για την οποία απευθύνεται, εξαρτήσεις που χρειάζεται για τη μεταγλώττισή του και τη σωστή λειτουργία του.

Το αρχείο debian/copyright περιέχει το κείμενο με τα πνευματικά δικαιώματα της εφαρμογής και την άδεια διανομής της.

Το αρχείο debian/rules είναι ένα αρχείο script το οποίο πληροφορεί το εργαλείο που έχει αναλάβει την κατασκευή του πακέτου για την ενέργεια που πρέπει να εκτελέσει, δηλαδή την κατασκευή, εγκατάσταση και άλλα. Αν και μπορεί τεχνικά να είναι ένα οποιουδήποτε τύπου αρχείο script, πάντα έχει τη μορφή Makefile.

Τα σύγχρονα γραφικά περιβάλλοντα εργασίας του Linux πλέον διαβάζουν τα αρχεία .desktop για την κατασκευή του μενού με τις εφαρμογές του συστήματος. Ωστόσο, στη διανομή Debian υπάρχει από παλιά ένα επιπλέον μενού και κάθε εφαρμογή για να περιλαμβάνεται σε αυτό το μενού πρέπει να έχει στο πακέτο της το αρχείο debian/menu. Αυτό το αρχείο περιέχει πληροφορίες όπως τον τίτλο της εφαρμογής, την εντολή που θα εκτελεστεί, το εικονίδιό της και την κατηγορία στην οποία υπάγεται.

Μετά την παραγωγή του πακέτου πρέπει να ελεγχθεί για τυχόν παραβιάσεις και σφάλματα. Αυτός ο έλεγχος συνήθως γίνεται με το εργαλείο lintian. Για να μπορέσει να ενσωματωθεί κάποιο πακέτο στα αποθετήρια της διανομής πρέπει οπωσδήποτε να επιτύχει στους ελέγχους που κάνει το εργαλείο lintian.

4.2.2. Η διανομή Gentoo

Λογότυπο διανομής Gentoo

Το 1999 ο Daniel Robbins δημιούργησε μια διανομή linux βασισμένη στην μεταγλώττιση από πηγαίο κώδικα. Το όνομά της τότε ήταν Enoch Linux και το 2002 μετονομάστηκε σε Gentoo. Το όνομα της διανομής προήλθε από ένα είδος πιγκουίνου, που είναι οι γρηγορότεροι κολυμβητές, γιατί παρείχε τα γρηγορότερα εκτελέσιμα και τον πιο γρήγορο (πειραγμένο) μεταγλωττιστή gcc. Το αρχικό target group στο οποίο στόχευε η διανομή ήταν οι προγραμματιστές και οι λεγόμενοι power users, δηλαδή όσοι έχτιζαν συχνά τα πακέτα τους από τον πηγαίο κώδικα και ήξεραν από πρώτο χέρι τα πλεονεκτήματα και τα μειονεκτήματα αυτής της επίπονης διαδικασίας.

Η βασική φιλοσοφία της διανομής Gentoo είναι η δημιουργία ενός λειτουργικού συστήματος που θα δουλεύει όπως θέλει ο χρήστης, και όχι να δουλεύει ο χρήστης όπως θέλει το λειτουργικό του. Ο χρήστης να εκμεταλλεύεται όλες τις δυνατότητες του software και του hardware και τα τελικά εκτελέσιμα να είναι κομμένα και ραμμένα στις ανάγκες του.

Μερικές από τις καινοτομίες που έφερε η διανομή Gentoo στον κόσμο του Linux, οι οποίες παραμένουν μέχρι και σήμερα ανάμεσα στα σημαντικότερα πλεονεκτήματα της διανομής έναντι των άλλων, είναι :

- Η μεγάλη τροποποίηση στον πηγαίο κώδικα του gcc. Όπως είναι φυσικό, είναι θέμα πρώτης προτεραιότητας, για μια διανομή που βασίζεται σε πακέτα πηγαίου κώδικα, η σωστή και γρήγορη λειτουργία του compiler
- Η πλήρης ευελιξία και το πλήθος των επιλογών. Κάθε χρήστης έχει τη δυνατότητα ακόμα και από το στάδιο της εγκατάστασης να τροποποιήσει τα πάντα. Δεν υπάρχει καμία προεπιλογή στο πώς και ποια πακέτα θα μεταγλωττιστούν για το σύστημα.
- Η πλήρης υποστήριξη για "εξωτικές" αρχιτεκτονικές. Σχεδόν από την αρχή της δημιουργίας του project υπήρχε ενδιαφέρον από διάφορους χρήστες για την εισαγωγή της διανομής σε μηχανήματα PPC και Sparc, μεταξύ άλλων. Φυσικά, η μεταγλώττιση από πηγαίο κώδικα καθιστούσε εύκολη αυτή τη μετάβαση. Αξίζει να αναφερθεί ότι το Gentoo υπήρξε η πρώτη διανομή (αλλά και το πρώτο λειτουργικό σύστημα) η οποία υποστήριξε την αρχιτεκτονική 64-bit (amd64).

- Η εκμετάλλευση στο έπακρο των δυνατοτήτων του εκάστοτε μηχανήματος. Η γενικότερη φιλοσοφία της διανομής να "παίζει" με διάφορες επιλογές των πακέτων και του μεταγλωττιστή δίνουν μία μοναδική ευκαιρία στον χρήστη να αξιοποιήσει πλήρως και με τον καλύτερο τρόπο όλους τους διαθέσιμους πόρους του συστήματος.
- Η εύκολη συγγραφή πακέτων (ebuilds) και η τροποποίησή τους από τους χρήστες.

4.2.2.1. Πακέτα Ebuild και Ebuild Policy

Το σύστημα πακέτων του Gentoo είναι το Portage, το οποίο αποτελείται από δύο βασικά μέρη, το Portage Tree με τα Ebuilds και το emerge. Ένα ebuild είναι ένα script που περιέχει πληροφορίες για κάποιο πακέτο, τις οποίες στέλνει στο Portage για την εγκατάσταση του πακέτου.. Οι πληροφορίες αυτές περιλαμβάνουν το όνομα του πακέτου, την ιστοσελίδα που βρίσκεται για να κατέβει, την άδεια με την οποία συνοδεύεται, και στη συνέχεια ενδεχόμενες επιπλέον ενέργειες που πρέπει να γίνουν πριν ή μετά την εξαγωγή του κώδικα, τη μεταγλώττιση του και την τελική εγκατάσταση. Επίσης, περιγράφει τις εξαρτήσεις που έχει το συγκεκριμένο πακέτο και τα USE flags που μπορεί να πάρει. Το emerge είναι το εργαλείο το οποίο χρησιμοποιεί ο χρήστης για να εγκαταστήσει, απεγκαταστήσει και αναβαθμίσει πακέτα στο σύστημά του. Αυτό λειτουργεί διαβάζοντας τα αρχεία ebuilds και εκτελώντας τις λειτουργίες που αυτά περιγράφουν.

Ο διαχειριστής πακέτου Portage είναι γραμμένος σε Python, ενώ τα ebuilds είναι γραμμένα σε bash, επιτρέποντας τη συγγραφή εντολών σε ένα ebuild που μπορούν γραφτούν και σε μια κονσόλα. Η μεταγλώττιση ενός ebuild γίνεται με τις εντολές ebuild και emerge. Η πρώτη αποτελεί μια εντολή που απλά μας επιτρέπει να εγκαταστήσουμε ένα ebuild, χωρίς όμως να κάνει επιπλέον ενέργειες όσον αφορά τις εξαρτήσεις. Η εντολή emerge μπορεί να περιγραφεί ως μια υψηλού επιπέδου μηχανή η οποία εγκαθιστά, παραμετροποιεί το πακέτο και λύνει το πρόβλημα των εξαρτήσεων.

Η ονομασία ενός ebuild αποτελείται από τέσσερις λογικές υποκατηγορίες:

pkg-ver{_suf{#}}{-r#}.ebuild

Το pkg είναι το όνομα του ebuild, που συνήθως είναι το ίδιο με το όνομα του πακέτου. Μπορεί να περιέχει μικρά γράμματα, αριθμούς και τους χαρακτήρες “-”, “_” και “+”. Το ver είναι η έκδοση του ebuild, που και πάλι πρέπει να είναι ίδια με την έκδοση του πακέτου. Συνήθως αποτελείται από δύο ή τρείς αριθμούς χωρισμένους με τελείες και ενδέχεται να υπάρχει ένα

μοναδικό γράμμα στο τέλος. Το υπόλοιπο του ονόματος είναι προαιρετικό και αφορά την προγραμματιστική κατάσταση του πακέτου, για παράδειγμα αν βρίσκεται σε κατάσταση alpha, beta, rc, ενώ το # αντικαθίσταται από έναν αριθμό με την έλλειψή του σημαίνει κανονική έκδοση . Το {-r#} δηλώνει την έκδοση του ebuild για το ίδιο πακέτο.

Ένα παράδειγμα αρχείου ebuild είναι το παρακάτω, όπως χρησιμοποιήθηκε για την εφαρμογή Giortes :

```
# Copyright 1999-2010 Gentoo Foundation
# Distributed under the terms of the GNU General Public License v2
# $Header: $

EAPI="2"
inherit autotools
DESCRIPTION="Displays the name day that is celebrated"
HOMEPAGE="http://linuxteam.cs.teilar.gr/~forfolias/giortes"
SRC_URI="http://linuxteam.cs.teialr.gr/~forfolias/${PN}/files/src/${P}.tar.gz"
LICENSE="GPL-3"
SLOT="0"
KEYWORDS="~amd64 ~x86"
IUSE="gtk nls"
RDEPEND="gtk? ( >=x11-libs/gtk+-2.16 )"
DEPEND="dev-util/pkgconfig
        nls? ( dev-util/intltool
                sys-devel/gettext )"

src_configure() {
        econf  $(use_enable gtk gtk-gui)
}
src_install() {
        emake DESTDIR="${D}" install || die "emake install failed"
        dodoc AUTHORS ChangeLog NEWS README
}
```

Εύκολα διακρίνεται πως ένα ebuild χωρίζεται σε δύο τμήματα. Στο τμήμα που περιέχει πληροφορίες για το πακέτο και το τμήμα του κώδικα που αφορά τη διαδικασία της εξαγωγής, μεταγλώττισης και εγκατάστασης του πακέτου. Οι πρώτες τρεις γραμμές αποτελούν την επικεφαλίδα που είναι η ίδια για όλα τα ebuilds και είναι απαραίτητη. Η επόμενη γραμμή δείχνει σε πιο ebuild API είναι γραμμένο το ebuild, με την πιο πρόσφατη έκδοση EAPI να είναι η 2. Στη συνέχεια υπάρχει η δήλωση κάποιων μεταβλητών.

Το SLOT είναι η θέση που θα μπει το πακέτο. Με αυτό εννοούμε ότι το portage επιτρέπει να υπάρχουν διαφορετικές εκδόσεις του ίδιου πακέτου εγκατεστημένες στο σύστημα, αρκεί να δηλωθούν με διαφορετικό SLOT η κάθε μία. Πρέπει όμως να εξασφαλίζεται ότι τα δύο πακέτα δε θα κάνουν εγκατάσταση αρχεία με ίδιο όνομα, ή τουλάχιστον θα πρέπει τα αρχεία τους να εγκαθίστανται σε διαφορετικούς φακέλους.

Τα KEYWORDS αφορούν την αρχιτεκτονική στην οποία απευθύνεται το πακέτο και στην ωριμότητα του πακέτου στην κάθε αρχιτεκτονική. Οι αρχιτεκτονικές που αρχίζουν με ~ δηλώνουν ότι το πακέτο είναι ασταθές για τη συγκεκριμένη αρχιτεκτονική, αλλιώς είναι σταθερό. Αυτό σημαίνει ότι η ίδια έκδοση ενός πακέτου μπορεί να ανήκει στο stable tree μιας αρχιτεκτονικής και ταυτόχρονα να ανήκει στο unstable tree μιας άλλης. Η σταθεροποίηση ενός πακέτου γίνεται μόνο από τις architecture teams, αφού ζητηθεί από το δημιουργό του πακέτου, ενώ η πρόσθεση μιας unstable αρχιτεκτονικής μπορεί να γίνει και από τον ίδιο το δημιουργό.

Το IUSE δηλώνει τα διαθέσιμα USE flags που θα έχει το ebuild και ανάλογα με τις επιλογές του χρήστη θα είναι ενεργοποιημένα ή απενεργοποιημένα. Στις μεταβλητές RDEPEND και DEPEND δηλώνονται οι εξαρτήσεις του πακέτου. Η μεταβλητή DEPEND είναι για τις εξαρτήσεις που απαιτούνται για τη μεταγλώττιση του πακέτου, και η μεταβλητή RDEPEND είναι για τη δήλωση εξαρτήσεων που απαιτούνται για το τρέξιμο και τη σωστή χρήση του πακέτου. Η εύρεση των εξαρτήσεων γίνεται με διάφορους τρόπους. Οι προφανείς είναι ο έλεγχος για τυχόν αναφορές στην ιστοσελίδα του πακέτου, σε αρχεία README και η ανάγνωση του σφάλματος μετά από αποτυχία εγκατάστασης λόγω έλλειψης εξάρτησης.

Κάποιες επιπλέον πολύ βασικές μεταβλητές που χρησιμοποιούνται είναι η P, η οποία επιστρέφει το όνομα και την έκδοση του πακέτου, η PN η οποία επιστρέφει μόνο το όνομα του πακέτου και η D η οποία επιστρέφει τον προσωρινό κατάλογο στον οποίο γίνεται επεξεργασία του πακέτου.

Υπάρχουν επίσης διάφορες συναρτήσεις οι οποίες επιδρούν στην εγκατάσταση του πακέτου. Στο παραπάνω παράδειγμα χρησιμοποιούνται δύο πολύ δημοφιλείς συναρτήσεις. Εφόσον

το πακέτο χρησιμοποιεί το GNU Build System, η διαδικασία εγκατάστασης του πακέτου με το χέρι θα ήταν αρχικά η εκτέλεση της εντολής “./configure ”, έπειτα η make και τέλος η make install . Η συνάρτηση src_configure αναλαμβάνει να εκτελέσει τις δύο πρώτες εντολές, με τη βοήθεια δύο βοηθητικών εντολών, στην πραγματικότητα είναι και αυτές συναρτήσεις, των econf και emake. Το econf υποστηρίζει το διακόπτη --enable-gtk-gui ή --disable-gtk-gui ανάλογα αν ο χρήστης έχει επιλεγμένο το gtk USE flag. Παρομοίως και η συνάρτηση src_install αναλαμβάνει την εγκατάσταση του πακέτου. Μια ακόμα χρήσιμη συνάρτηση είναι η die η οποία διακόπτει την τρέχουσα λειτουργία αν αυτή αποτύχει και επιστρέφει ένα μήνυμα που δίνεται μετά από αυτήν .

Αναφορές

http://hellug.gr/

http://debian.gr/

http://d-i.alioth.debian.org/manual/el.i386/index.html

http://www.gnu.org/philosophy/microsoft-antitrust.html

http://www.mingw.org/wiki/

http://el.wikipedia.org/wiki/Linux

http://en.wikipedia.org/wiki/Windows

http://en.wikipedia.org/wiki/MinGW

http://nsis.sourceforge.net/Docs/

http://gtk-win.sourceforge.net/home/index.php/en/Embedding

http://alioth.debian.org/docman/view.php/30046/2/menu-one-file.html

http://www.debian.org/doc/maint-guide/

http://www.debian.org/doc/debian-policy/

http://devmanual.gentoo.org/

http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml?part=3&chap=1

ΚΕΦΑΛΑΙΟ 5º

Το ελεύθερο λογισμικό Giortes

Στα πλαίσια αυτής της πτυχιακής εργασίας δημιουργήθηκε ένα έργο ελεύθερου λογισμικού με το όνομα Giortes. Όπως μαρτυρά και το όνομα της εφαρμογής, πρόκειται για ένα πρόγραμμα εορτολογίου. Είναι μια εφαρμογή με σκοπό να ενημερώνει καθημερινά με έναν πολύ πρακτικό τρόπο τους χρήστες για τις επερχόμενες γιορτές και άλλα γεγονότα. Η εφαρμογή μπορεί να φανεί ιδιαίτερα χρήσιμη στους Έλληνες χρήστες Linux και υποστηρικτές του ελεύθερου λογισμικού, καθώς αποτελεί μια από τις ελάχιστες ελεύθερες υλοποιήσεις σε τέτοιου είδους λογισμικό. Ακόμη, προσφέρει μεγάλη ευελιξία λόγω του τρόπου σχεδιασμού της, των δυνατοτήτων και της ευχρηστίας που διαθέτει.

Η εφαρμογή προσφέρει λειτουργίες αναζήτησης ονομάτων και υπολογισμού της Κυριακής του Πάσχα και των αργιών ενός έτους. Ακόμη, δίνει τη δυνατότητα στους χρήστες να προσθέσουν δικά τους γεγονότα προς εμφάνιση με έναν εύκολο τρόπο, προσθέτοντας την ημερομηνία και τον τίτλο του γεγονότος σε ένα αρχείο. Τέλος, παρέχει μια λίστα με εορτασμούς παγκόσμιων ημερών, όπως για παράδειγμα η παγκόσμια ημέρα κατά του AIDS, την οποία μπορεί να επεκτείνει ο χρήστης. Η εφαρμογή διατίθεται κάτω από τη Γενική Άδεια Δημόσιας Χρήσης GNU της έκδοσης 3.

5.1. Περιληπτική λειτουργία

Ο κώδικας της εφαρμογής είναι χωρισμένος σε πολλά αρχεία και συναρτήσεις με σαφή ονόματα και εύκολες μικρές λειτουργίες. Μόλις το πρόγραμμα ξεκινήσει και μπει στη συνάρτηση main(), αρχικά διαπιστώνεται η γλώσσα του συστήματος ώστε να χρησιμοποιηθεί η κατάλληλη μετάφραση, εάν αυτή παρέχεται. Στη συνέχεια καλείται η συνάρτηση Initialize() στην οποία γίνονται κάποιες αρχικοποιήσεις τιμών, διαπιστώνεται η ημερομηνία του συστήματος, υπολογίζεται η Κυριακή του Πάσχα για το παρόν έτος και διαβάζεται το αρχείο με τις ρυθμίσεις του χρήστη.

Εφόσον γίνουν οι κατάλληλες αρχικοποιήσεις, η εφαρμογή ελέγχει για τυχόν ορίσματα που έχουν δοθεί από τη γραμμή εντολών και αντικαθιστά κατάλληλα τις επιλογές που είχε διαβάσει από το αρχείο ρυθμίσεων. Στη συνέχεια, μέσω των συναρτήσεων checkFile(), checkPersonal() και checkInternationalDays() ελέγχεται η ύπαρξη του αρχείου με τις εορτές, του αρχείου με τα προσωπικά γεγονότα και η λίστα με τους εορτασμούς παγκόσμιων ημερών. Τέλος, καλείται η συνάρτηση checkAllDays(), η οποία διαβάζει τα παραπάνω αρχεία για τον αριθμό των ημερών που ορίστηκε στο αρχείο ρυθμίσεων, και καλεί τη συνάρτηση αναζήτησης find() με τις κατάλληλες ημερομηνίες προς αναζήτηση ώστε να εμφανιστούν τα γεγονότα.

Εικόνα 5.1: Το παράθυρο με τις πληροφορίες της εφαρμογής

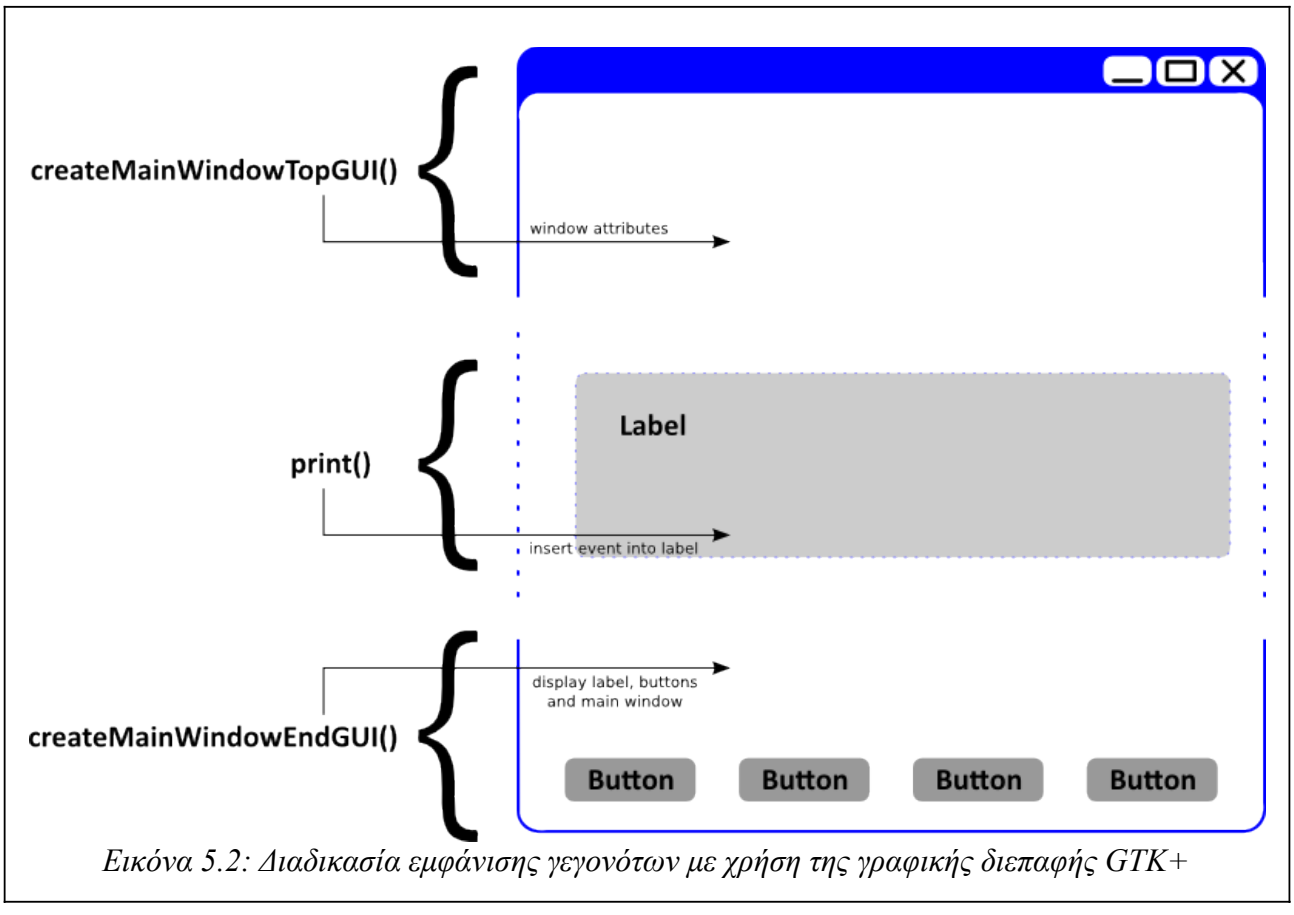
5.2. CLI/GUI version

Η εφαρμογή Giortes μπορεί να λειτουργήσει με δυο διεπαφές, αυτή της γραμμής εντολών και αυτή με τη βιβλιοθήκη GTK+. Οι συναρτήσεις που αφορούν τη γραφική διεπαφή βρίσκονται στο αρχείο interface.c. Η εφαρμογή έχει δημιουργηθεί με γνώμονα την ελευθερία παραμετροποίησης και επέκτασης οπότε σχετικά εύκολα μπορεί να χρησιμοποιηθεί κάποια διαφορετική γραφική διεπαφή από τη GTK+. Η επιλογή της γραφικής διεπαφής ή όχι γίνεται κατά τη διαδικασία μεταγλώττισης παρέχοντας το όρισμα --enable-gtk-gui στο configure script που αναλαμβάνει να προετοιμάσει την εφαρμογή για το σύστημα.

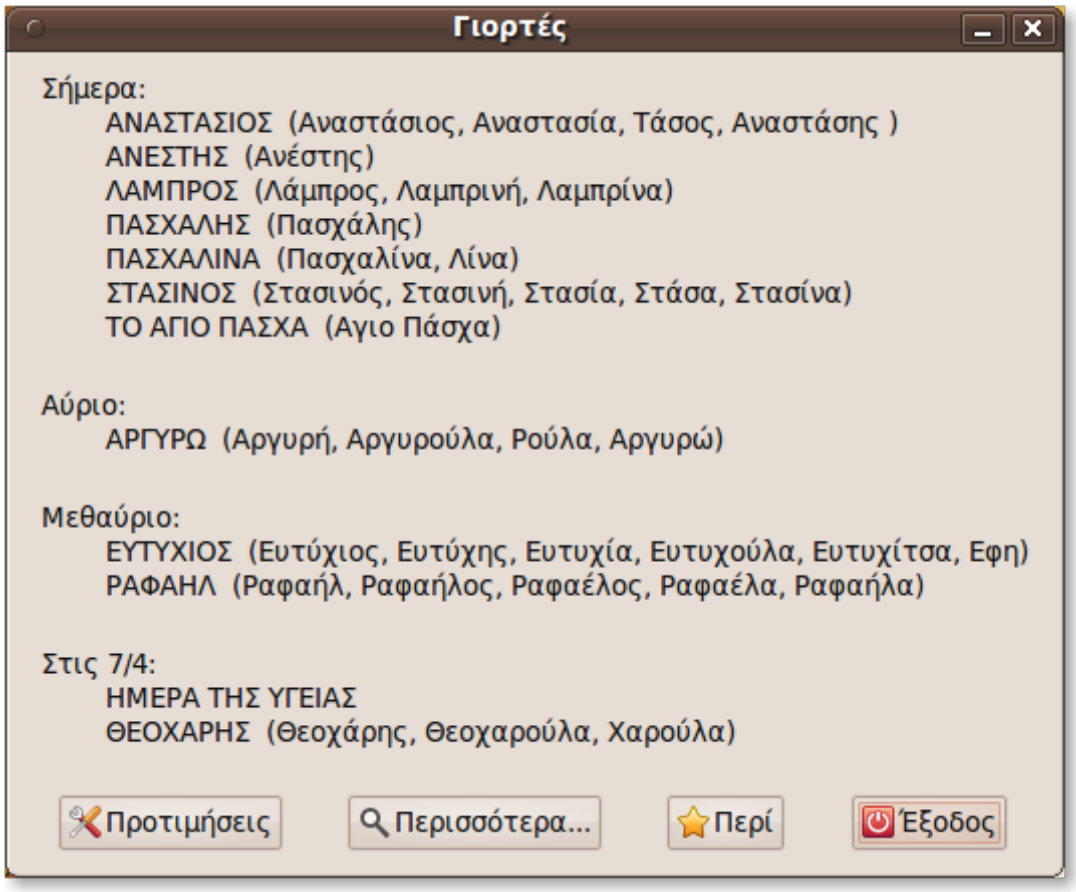
Όταν δε γίνεται χρήση της γραφικής διεπαφής τα γεγονότα απλά εμφανίζονται στην οθόνη του τερματικού κάνοντας χρήση της συνάρτησης fprintf(). Όταν όμως η εφαρμογή έχει μεταγλωττιστεί ώστε να χρησιμοποιεί γραφική διεπαφή, η διαδικασία εμφάνισης των γεγονότων είναι διαφορετική. Υπάρχουν στο αρχείο interface.h κάποιες global μεταβλητές οι οποίες χρησιμοποιούνται για το κυρίως παράθυρο της εφαρμογής, το εικονίδιό της και τα περιεχόμενα

προς εμφάνιση. Πιο συγκεκριμένα η char label_str κρατάει το κείμενο από τα γεγονότα προς εμφάνιση το οποίο χρησιμοποιείται από τη GtkWidget *label για την εμφάνισή τους στο κυρίως παράθυρο.

Καθώς ξεκινά η εφαρμογή δημιουργείται με τη συνάρτηση createMainWindowTopGUI() το κυρίως παράθυρό της, ορίζοντας τον τίτλο που θα έχει, το εικονίδιο που θα χρησιμοποιεί, τις διαστάσεις του και άλλα παρόμοια χαρακτηριστικά. Ωστόσο το παράθυρο δεν εμφανίζεται ακόμη στην οθόνη του χρήστη. Κατά την αναζήτηση των γεγονότων, γεμίζει η μεταβλητή label_str με το κατάλληλο κείμενο των γεγονότων και στη συνέχεια τη χρησιμοποιεί η label για να τα ενσωματώσει στο κυρίως παράθυρο. Αυτό γίνεται για κάθε μέρα προς εμφάνιση. Τέλος, εφόσον έχει τελειώσει η αναζήτηση και η ενσωμάτωση των γεγονότων, καλείται η συνάρτηση createMainWindowEndGUI() η οποία προσθέτει τα κουμπιά στο παράθυρο και ειδοποιεί την GTK+ ότι το παράθυρο είναι έτοιμο προς εμφάνιση.



Εικόνα 5.2: Διαδικασία εμφάνισης γεγονότων με χρήση της γραφικής διεπαφής GTK+



Εικόνα 5.3: Το κυρίως παράθυρο της εφαρμογής

5.3. Command line arguments

Η εφαρμογή Giortes μπορεί να καλέσει κάποια λειτουργία της κατευθείαν περνώντας της το κατάλληλο όρισμα από τη γραμμή εντολών. Πιο αναλυτικά τα ορίσματα που δέχεται είναι :

- -h : εμφανίζει ένα κείμενο βοήθειας, πληροφορίες επικοινωνίας και συνοπτικά την άδεια χρήσης της εφαρμογής.
- -v : εμφανίζει την έκδοση της εφαρμογής.
- -e : ζητά από το χρήστη να εισάγει τον αριθμό ενός έτους και στη συνέχεια υπολογίζει και εμφανίζει την Κυριακή του Πάσχα για αυτό το έτος.
- -a : ζητά από το χρήστη να εισάγει τον αριθμό ενός έτους και στη συνέχεια υπολογίζει και εμφανίζει τις επίσημες αργίες για αυτό το έτος.
- -q : αποκρύπτει από τον χρήστη μηνύματα λάθους

- -n : δέχεται σαν επιπλέον όρισμα κάποιο από τα γράμματα a, p, i, n και h. Με το γράμμα a αποκρύπτει από την εμφάνιση τα γεγονότα που έχει ορίσει ο χρήστης και τους εορτασμούς παγκόσμιων ημερών. Με το γράμμα p αποκρύπτει μόνο τα γεγονότα του χρήστη ενώ με το i μόνο τους παγκόσμιους εορτασμούς. Με το γράμμα n, το οποίο είναι και η προεπιλογή για την εφαρμογή, δεν αποκρύπτει τίποτα, δηλαδή εμφανίζει όλα τα γεγονότα και τέλος με το h εμφανίζει ένα κείμενο βοήθειας του ορίσματος -n.

- -m : δέχεται σαν επιπλέον όρισμα κάποιο από τα γράμματα a, p, i, h. Με το γράμμα a δημιουργεί ένα πρότυπο αρχείο για τα προσωπικά γεγονότα του χρήστη και το αρχείο με τους εορτασμούς παγκόσμιων ημερών. Με το γράμμα p δημιουργεί μόνο το αρχείο με τα γεγονότα του χρήστη, με το i μόνο τους παγκόσμιους εορτασμούς και με το h εμφανίζει ένα κείμενο βοήθειας του ορίσματος -m.

- -x : δέχεται σαν επιπλέον όρισμα έναν ακέραιο αριθμό ο οποίος συμβολίζει τις επόμενες ημέρες για τις οποίες θα πρέπει να εμφανίσει γεγονότα. Σε περίπτωση αρνητικού αριθμού χρησιμοποιείται ο αριθμός 5, ο οποίος είναι ο προεπιλεγμένος. Σε περίπτωση αριθμού μεγαλύτερου από το 365, δηλαδή ενός έτους, αφαιρείται από αυτόν συνεχώς το 365 μέχρι να φτάσει σε έναν αριθμό μικρότερο του 365.

- -d : δέχεται σαν επιπλέον όρισμα τη διαδρομή του αρχείου που θα χρησιμοποιηθεί για την αναζήτηση των εορτών.

- -p : δέχεται σαν επιπλέον όρισμα τη διαδρομή του αρχείου που θα χρησιμοποιηθεί ως το αρχείο με τα προσωπικά γεγονότα του χρήστη.

- -i : δέχεται σαν επιπλέον όρισμα τη διαδρομή του αρχείου που θα χρησιμοποιηθεί ως το αρχείο με τους εορτασμούς παγκόσμιων ημερών.

- -w : γράφει τις ρυθμίσεις που έχουν προηγηθεί αυτού του ορίσματος στο αρχείο με τις ρυθμίσεις του χρήστη.

- -s : ζητά από το χρήστη ένα αλφαριθμητικό για το οποίο θα πραγματοποιήσει αναζήτηση.

- -f : ζητά από το χρήστη μια ημερομηνία για την οποία θα πραγματοποιήσει αναζήτηση.

- -c : δέχεται σαν επιπλέον όρισμα έναν ακέραιο αριθμό ο οποίος συμβολίζει τα δευτερόλεπτα για τα οποία θα εμφανιστεί το παράθυρο της εφαρμογής πριν κλείσει αυτόματα. Αυτό το

όρισμα υπάρχει μόνο αν γίνει μεταγλώττιση της εφαρμογής με χρήση της γραφικής διεπαφής. Από προεπιλογής αυτός ο αριθμός είναι το 30.

5.4. Αρχείο ονομαστικών εορτών

Το αρχείο των ονομαστικών εορτών είναι το κύριο αρχείο της εφαρμογής. Αν αυτό το αρχείο δε βρεθεί από την εφαρμογή, τότε αυτή τερματίζει με κάποιο μήνυμα λάθους. Πρόκειται για ένα αρχείο δυαδικής μορφής που ορίζεται στο header file init.h και περιέχει εγγραφές της μορφή που έχει η struct event :

```
struct event {
   int mera;
   int minas;
   char kiniti[4];
   char onoma[200];
   char sxolia[400];
};
```

Οι πρώτες δύο μεταβλητές που ορίζονται στη struct event, η mera και minas, αφορούν όπως φανερώνει και το όνομά τους, την ημέρα και το μήνα της εορτής ή του γεγονότος. Στη συνέχεια υπάρχει η μεταβλητή kiniti η οποία χρησιμοποιείται για να διαπιστωθεί αν κάποια εορτή είναι κινητή, δηλαδή αν εξαρτάται από την Κυριακή του Πάσχα. Αν η εορτή δεν εξαρτάται από την Κυριακή του Πάσχα τότε περιέχει το χαρακτήρα “*”, ενώ σε αντίθετη περίπτωση περιέχει τον αριθμό από τον οποίο εξαρτάται. Για παράδειγμα μπορεί να περιέχει τον αριθμό -2 που δηλώνει ότι το γεγονός πρέπει να εμφανιστεί 2 ημέρες πριν την Κυριακή του Πάσχα. Ακόμη, επειδή υπάρχουν κάποιες κινητές εορτές οι οποίες δεν εξαρτώνται από το Πάσχα αλλά από άλλα γεγονότα, η μεταβλητή kiniti για αυτές θα περιέχει μια τιμή διαφορετική από το “*” και διαχειρίζονται με βάση το όνομά τους από τη συνάρτηση checkKiniti(). Οι τελευταίες δυο μεταβλητές της struct event χρησιμοποιούνται για το όνομα της εορτής και επιπλέον πληροφορίες ή σχόλια.

Χρησιμοποιήθηκε αρχείο δυαδικής μορφής κυρίως για να είναι σχετικά δύσκολο στον χρήστη να το αλλάξει, εφόσον είναι αρχείο ζωτικής σημασίας για την εφαρμογή. Ακόμη, επειδή είναι δυνατή η άμεση προσπέλαση μιας εγγραφής χωρίς να χρειάζεται να προσπελαστούν σειριακά όλες οι προηγούμενες. Τέλος, είναι σαφώς ταχύτερη η διαχείρισή τους σε σχέση με τα κανονικά αρχεία κειμένου γιατί μια ψηφιακή εικόνα της εγγραφής μεταφέρεται από τη μνήμη στο δίσκο και αντίστροφα, ενώ στα αρχεία κειμένου πρέπει τα δεδομένα να μετατρέπονται σε κείμενο είτε γράφονται στη μνήμη είτε στο δίσκο και αυτό καθυστερεί.

5.5. Αρχεία προσωπικών γεγονότων και παγκόσμιων ημερών

Εκτός από το κύριο αρχείο με τις εορτές το πρόγραμμα Giortes υποστηρίζει κάποια επιπλέον αρχεία, τα οποία χρησιμοποιούνται για να εμφανίσουν γεγονότα που έχει ορίσει ο χρήστης και εορτασμούς παγκόσμιων ημερών. Αυτά τα αρχεία είναι σε μορφή απλού κειμένου ώστε να μπορούν να επεξεργαστούν και να επεκταθούν εύκολα από το χρήστη. Ακόμη, μπορούν να παραχθούν από την εφαρμογή εάν ο χρήστης τα διαγράψει, είτε με χρήση του ορίσματος -m της γραμμής εντολών, είτε μέσα από τις επιλογές του προγράμματος.

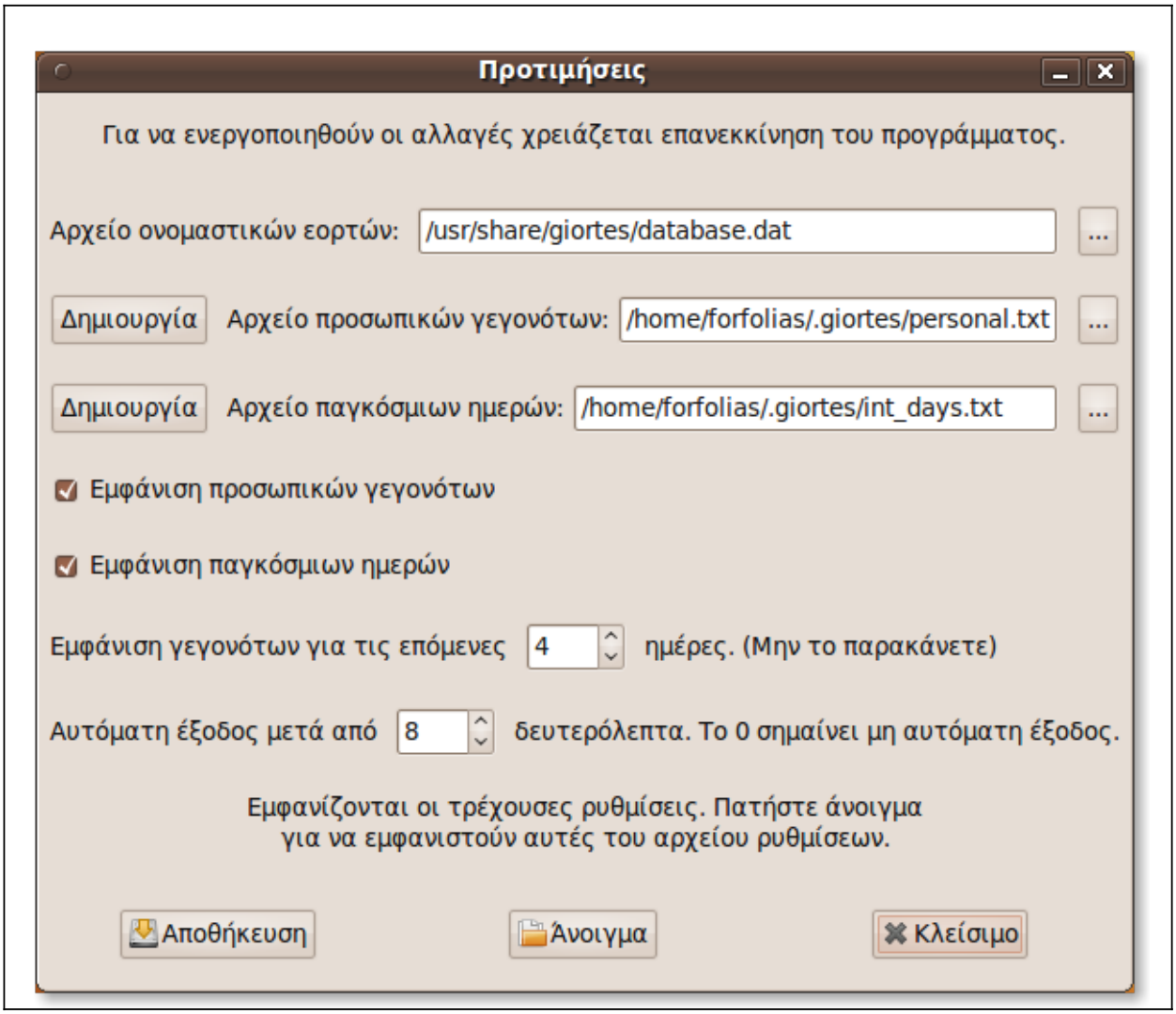
Αυτά τα αρχεία ακολουθούν μια συγκεκριμένη δομή ώστε να μπορεί να τα διαβάζει η εφαρμογή. Αυτή η δομή είναι πρώτα να αναγράφεται ο αριθμός της ημέρας του γεγονότος, έπειτα ο μήνας και τέλος το μήνυμα προς εμφάνιση. Για παράδειγμα αν σε κάποιο από αυτά τα αρχεία υπάρχει η εγγραφή “23 05 Γενέθλια Γιώργου”, τότε στις 23 Μαΐου η εφαρμογή θα εμφανίσει επιπλέον το μήνυμα “Γενέθλια Γιώργου”. Ακόμη, αν κάποια γραμμή ξεκινά με τον χαρακτήρα “#” δεν λαμβάνεται υπόψιν από την εφαρμογή καθώς θεωρείται σχόλιο.

Τα αρχεία αυτά στο λειτουργικό σύστημα Linux είναι διαφορετικά για κάθε χρήστη του συστήματος και αποθηκεύονται σε έναν κρυφό κατάλογο με όνομα .giortes στο home του κάθε χρήστη. Στο λειτουργικό σύστημα Windows, τα αρχεία βρίσκονται στο φάκελο giortes στον κατάλογο του συστήματος Program Files, εκεί δηλαδή που εγκαθίσταται η εφαρμογή. Το αρχείο με τα προσωπικά γεγονότα από προεπιλογής ονομάζεται personal.txt, ενώ το αρχείο με τους παγκόσμιους εορτασμούς int_days.txt.

5.6. Αρχείο ρυθμίσεων

Η εφαρμογή Giortes υποστηρίζει ένα αρχείο το οποίο περιλαμβάνει επιλογές του χρήστη ώστε να παραβλέπονται οι προεπιλεγμένες ρυθμίσεις. Το αρχείο αυτό είναι επίσης ένα απλό αρχείο κειμένου και βρίσκεται στον ίδιο κατάλογο με τα αρχεία προσωπικών γεγονότων και παγκόσμιων ημερών, ενώ έχει το όνομα settings.txt. Όταν εκτελεστεί η εφαρμογή για πρώτη φορά, το αρχείο αυτό θα δημιουργηθεί αυτόματα με τις προεπιλεγμένες τιμές των ρυθμίσεων.

Ακολουθεί κι αυτό με τη σειρά του μια συγκεκριμένη δομή. Αρχικά αναγράφεται το όνομα της ρύθμισης, ακολουθεί μια άνω-κάτω τελεία και στο τέλος είναι η τιμή που θα έχει η ρύθμιση μέσα σε εισαγωγικά. Αν υπάρχει στο αρχείο μια ρύθμιση την οποία δεν κατανοεί η εφαρμογή, τότε αυτή θα παραβλεφθεί. Ακόμη, αν κάποια γραμμή ξεκινά με τον χαρακτήρα “#” δεν λαμβάνεται υπόψιν από την εφαρμογή καθώς θεωρείται σχόλιο.



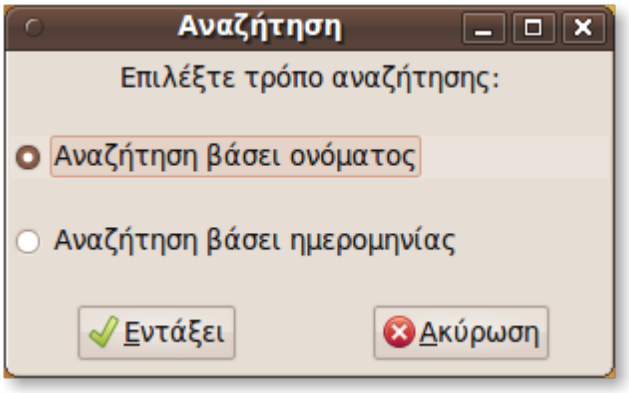
Οι ρυθμίσεις τις οποίες αντιλαμβάνεται η εφαρμογή είναι :

- show_X_next_days : για τον αριθμό των επόμενων ημερών προς εμφάνιση. Λειτουργεί σαν να έχει δοθεί το όρισμα -x
- name_database_file : για τη διαδρομή του αρχείου με τις εορτές. Λειτουργεί σαν να έχει δοθεί το όρισμα -d
- personal_database_file : για τη διαδρομή του αρχείου με τα προσωπικά γεγονότα. Λειτουργεί σαν να έχει δοθεί το όρισμα -p
- international_days_database_file : για τη διαδρομή του αρχείου με τους εορτασμούς των παγκόσμιων ημερών. Λειτουργεί σαν να έχει δοθεί το όρισμα -i

- show_personal_database : για το αν θα εμφανίζονται γεγονότα από το αρχείο με τα προσωπικά γεγονότα. Αυτή η ρύθμιση παίρνει την τιμή TRUE ή FALSE. Λειτουργεί σαν να έχει δοθεί το όρισμα -n p
- show_international_days_database : για το αν θα εμφανίζονται γεγονότα από το αρχείο με τους εορτασμούς των παγκόσμιων ημερών. Αυτή η ρύθμιση παίρνει την τιμή TRUE ή FALSE. Λειτουργεί σαν να έχει δοθεί το όρισμα -n i
- quiet : για το αν θα αποκρύπτει από τον χρήστη μηνύματα λάθους. Λειτουργεί σαν να έχει δοθεί το όρισμα -q
- autoclose_after_X_seconds : για τον αριθμό των δευτερολέπτων για τα οποία θα εμφανίζεται το παράθυρο της εφαρμογής πριν κλείσει αυτόματα. Λειτουργεί σαν να έχει δοθεί το όρισμα -c

5.7. Λειτουργίες αναζήτησης

Το πρόγραμμα Giortes υποστηρίζει δυο λειτουργίες αναζήτησης, η μια εκ των οποίων χρησιμοποιείται συνεχώς μέσα στην εφαρμογή για την εύρεση και εκτύπωση των γεγονότων. Μέσω της συνάρτησης search() γίνεται αναζήτηση βάσει ονόματος, ενώ με τη συνάρτηση find() γίνεται αναζήτηση βάσει ημερομηνίας.

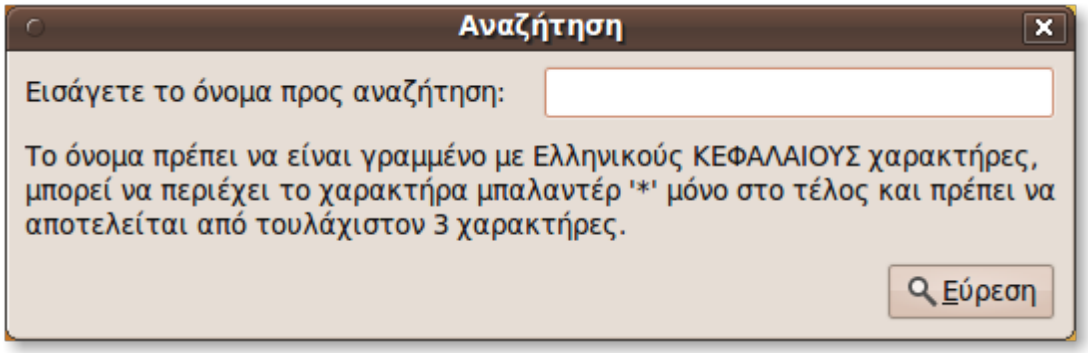


Εικόνα 5.5: Το παράθυρο επιλογής τύπου αναζήτησης

Η συνάρτηση search() δέχεται ένα όρισμα, το αλφαριθμητικό προς αναζήτηση. Αν το αλφαριθμητικό αυτό είναι μικρότερο από 3 χαρακτήρες τότε εμφανίζει κάποιο μήνυμα λάθους,

καθώς τα αποτελέσματα θα είναι πολλά και το παράθυρο θα γίνει τεράστιο. Το αλφαριθμητικό προς αναζήτηση μπορεί να περιέχει τον χαρακτήρα μπαλαντέρ “*” μόνο στο τέλος. Για παράδειγμα, αν δοθεί προς αναζήτηση το αλφαριθμητικό “ΜΑΡΙ*”, τα αποτελέσματα θα περιέχουν γιορτές από τα ονόματα ΜΑΡΙΟΣ, ΜΑΡΙΑ, ΜΑΡΙΑΝΝΑ κλπ.

Πιο αναλυτικά η παραπάνω συνάρτηση αρχικά ελέγχει αν το αλφαριθμητικό που δέχτηκε σαν όρισμα περιέχει το χαρακτήρα μπαλαντέρ και αν ναι σε ποια θέση. Στη συνέχεια, κάνει την αναζήτηση βάσει ονόματος στο αρχείο με τις εορτές και όταν υπάρχει κάποιο αποτέλεσμα καλείται η συνάρτηση checkKiniti(). Αυτό γίνεται για να ελεγχθεί αν η γιορτή είναι κινητή και να αλλάξει κατάλληλα ημερομηνία σύμφωνα με το τρέχον έτος. Τέλος καλείται η συνάρτηση print() που αναλαμβάνει την εκτύπωση της γιορτής στο τερματικό ή την ενσωμάτωσή της στο κυρίως παράθυρο. Με παρόμοιο τρόπο γίνεται στη συνέχεια και η αναζήτηση στα αρχεία με τα προσωπικά γεγονότα και τους παγκόσμιους εορτασμούς.

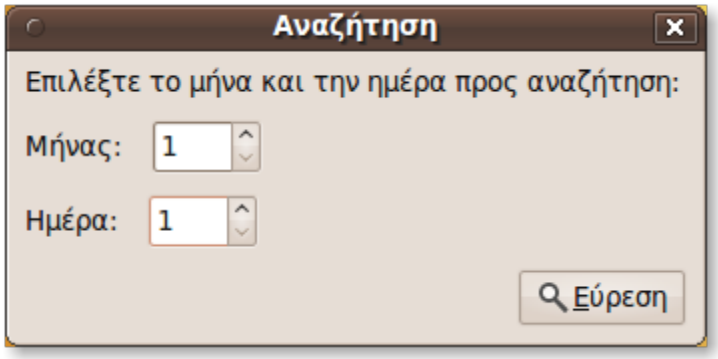


Εικόνα 5.6: Το παράθυρο εισαγωγής του ονόματος προς αναζήτηση

Η συνάρτηση find() κάνει αναζήτηση βάσει της τρέχουσας ημερομηνίας και δέχεται δυο ορίσματα. Το πρώτο όρισμα είναι μια σημαία για να διαπιστωθεί αν η συνάρτηση καλέστηκε από το χρήστη, οπότε σε περίπτωση μη εύρεσης αποτελεσμάτων να τον ειδοποιήσει με το κατάλληλο μήνυμα. Το δεύτερο όρισμα χρησιμοποιείται για να αλλάξει την τρέχουσα ημερομηνία αναζήτησης. Και τα δύο ορίσματα ουσιαστικά υπάρχουν για να διαφοροποιούν τη λειτουργία της αναλόγως αν καλέστηκε από τον χρήστη ή αυτόματα από την εφαρμογή για την κανονική προβολή των γεγονότων.

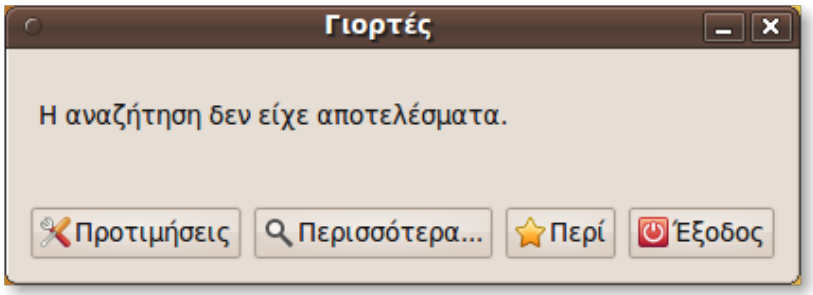
Αναλυτικότερα η συνάρτηση find() αρχικά ελέγχει το πρώτο της όρισμα και εάν αυτό είναι μηδέν σημαίνει ότι η συνάρτηση καλείται από τον χρήστη γιατί θέλει να κάνει αναζήτηση. Τότε τον

προτρέπει να εισάγει την ημερομηνία που αναζητά και ελέγχει αν αυτή η ημερομηνία είναι έγκυρη. Εάν η παραπάνω ημερομηνία δεν είναι έγκυρη θα εμφανιστεί ένα μήνυμα λάθους και ο χρήστης θα πρέπει να την εισάγει ξανά. Αν το πρώτο όρισμα της συνάρτησης είναι διάφορο του μηδενός σημαίνει ότι κλήθηκε εσωτερικά από την εφαρμογή για την αναζήτηση και προβολή κάποιου γεγονότος. Τότε, η ημερομηνία προς αναζήτηση αλλάζει σύμφωνα με το δεύτερο όρισμα και ελέγχεται αν είναι στα πλαίσια του μήνα ή θα πρέπει να αναζητηθεί στον επόμενο μήνα ή ακόμη και χρόνο.



Εικόνα 5.7: Το παράθυρο εισαγωγής της ημερομηνίας προς αναζήτηση

Εφόσον έχει τελειώσει ο ορισμός της ημερομηνίας, με τον ένα ή με τον άλλο τρόπο, ξεκινά η αναζήτηση αρχικά στο αρχείο με τα προσωπικά γεγονότα και έπειτα στο αρχείο με τους εορτασμούς των παγκόσμιων ημερών. Στη συνέχεια γίνεται αναζήτηση στο κυρίως αρχείο διαβάζοντας σειριακά όλες τις εορτές, ελέγχοντας αν είναι κινητές ή όχι, μέχρι δύο μήνες μετά την ημερομηνία προς αναζήτηση καθώς η μέγιστη εξάρτηση κινητής εορτής από το Πάσχα είναι 56 ημέρες αργότερα. Τέλος, αν η αναζήτηση είχε αποτελέσματα τότε αυτά εμφανίζονται στην οθόνη. Διαφορετικά, αν η συνάρτηση κλήθηκε από τον χρήστη ο τελευταίος ενημερώνεται με κάποιο μήνυμα για την αποτυχία της αναζήτησης, ενώ αν η συνάρτηση κλήθηκε από την ίδια την εφαρμογή δεν υπάρχει λόγος προειδοποίησής του.



Εικόνα 5.8: Το κυρίως παράθυρο πληροφορώντας ότι η αναζήτηση ήταν ανεπιτυχής

5.8. Αλγόριθμος υπολογισμού Κυριακής του Πάσχα

Το Πάσχα είναι κινητή γιορτή, η ημερομηνία εορτασμού του αλλάζει κάθε χρόνο. Όπως κινητές ημερολογιακά είναι και όλες οι άλλες γιορτές που συνδέονται μαζί του όπως η Κυριακή του Θωμά, η Πεντηκοστή, του Αγίου Πνεύματος και άλλες. Ο προσδιορισμός της Κυριακής του Πάσχα αποτέλεσε περίπλοκο πρόβλημα για τις Εκκλησίες και προκάλεσε σφοδρές έριδες. Εν τέλει, η Α' Οικουμενική Σύνοδος της Νίκαιας, το 325 μ.Χ., όρισε πως το χριστιανικό Πάσχα εορτάζεται την πρώτη Κυριακή μετά την Πανσέληνο, που θα γίνει κατά την ημέρα της εαρινής ισημερίας ή μετά από αυτήν. Αν η πανσέληνος γίνει Κυριακή τότε το Πάσχα θα εορτάζεται την αμέσως επομένη Κυριακή.

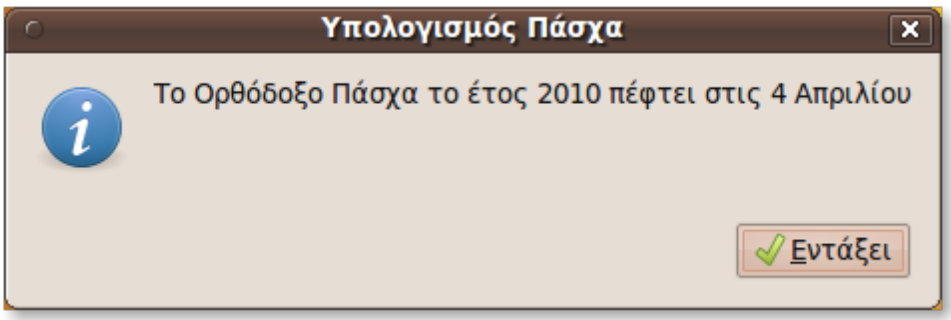
Επειδή λοιπόν η πρώτη μέρα της Σελήνης τοποθετείται μεταξύ 8ης Μαρτίου και 5ης Απριλίου, το Πάσχα μπορεί να πέσει στο διάστημα μεταξύ 22ας Μαρτίου και 25ης Απριλίου. Οι ημερομηνίες αυτές υπολογίζονται με το παλαιό ημερολόγιο γιατί μ' αυτό καθορίζεται η Εαρινή Ισημερία. Για να βρούμε τα όρια μέσα στα οποία γιορτάζεται το Πάσχα, σύμφωνα με το νέο ημερολόγιο, προσθέτουμε 13 μέρες, μέχρι το 2099, καθώς από τότε και μετά η διαφορά θα γίνει 14 ημέρες. Έτσι βρίσκουμε ότι οι ημερομηνίες του Ορθόδοξου Πάσχα κυμαίνονται από τις 4 Απριλίου ως τις 8 Μαΐου.

Για τον καθορισμό του Ορθόδοξου Πάσχα η Εαρινή Ισημερία υπολογίζεται με βάση το παλαιό (Ιουλιανό) ημερολόγιο, ενώ οι Ρωμαιοκαθολικοί και οι Προτεστάντες καθορίζουν την εαρινή ισημερία με βάση το νέο ημερολόγιο και γι' αυτό γιορτάζουν συνήθως το Πάσχα μια βδομάδα νωρίτερα από τους Ορθόδοξους. Κοινό Πάσχα έχουμε όταν η Γρηγοριανή και η Ιουλιανή πασχαλινή πανσέληνος πέσουν από την Κυριακή μέχρι και το Σάββατο της ίδιας βδομάδας, οπότε την αμέσως επόμενη Κυριακή έχουμε κοινό Πάσχα.

Ο Αθηναίος αστρονόμος Μέτων ανακάλυψε, το 432 π.Χ. ότι η περίοδος των 19 ηλιακών ετών ή 6940 ημερών περίπου, συγκροτούν έναν επαναλαμβανόμενο κύκλο κι αυτός είναι πρακτικά χρήσιμος, διότι αν καταγράψουμε τις ημερομηνίες των φάσεων της σελήνης επί 19 συνεχόμενα έτη, οι φάσεις θα επανέρχονται στις ίδιες ημερομηνίες και κατά την ίδια σειρά στα επόμενα 19 έτη κ.ο.κ. Η περίοδος αυτή ονομάστηκε Κύκλος του Μέτωνα, ο οποίος όμως παρουσιάζει μικρή απόκλιση, από το 325 μ.Χ. μέχρι σήμερα υπολογίζεται σε 5 περίπου ημέρες.

Με βάση λοιπόν τον Κύκλο του Μέτωνα, σχηματίσθηκε ο πίνακας των πανσελήνων του Πάσχα δηλαδή των μετά την 21η Μαρτίου Ιουλιανού Ημερολογίου πανσελήνων, που ακόμη και σήμερα χρησιμοποιούν οι Ορθόδοξοι Χριστιανοί. Αυτό συνεχίστηκε μέχρι το 1582 που η Καθολική Εκκλησία καθιέρωσε το νέο Γρηγοριανό Ημερολόγιο, που ισχύει σήμερα, για να διορθώσει το συσσωρευμένο λάθος του παλαιού Ιουλιανού Ημερολογίου.

Ο υπολογισμός της ημερομηνίας του Ορθόδοξου Πάσχα είναι ένα σύνθετο μαθηματικό θέμα και απαιτεί πρακτικά τις 4 πράξεις της αριθμητικής. Συνοπτικά η διαδικασία υπολογισμού είναι αρχικά από τον αριθμό του έτους, για το οποίο γίνεται ο προσδιορισμός, να αφαιρεθεί ο αριθμός 2. Έπειτα διαιρείται με τον αριθμό 19 και το υπόλοιπο της διαίρεσης πολλαπλασιάζεται με τον αριθμό 11. Στη συνέχεια, το γινόμενο του πολλαπλασιασμού διαιρείται με τον αριθμό 30, με το υπόλοιπο αυτής της διαίρεσης να λέγεται “Επακτή” και συμβολίζεται με το γράμμα “Ε”. Τέλος, πρέπει να αφαιρεθεί το “Ε” από τον αριθμό 44 και αν το «Ε» είναι μεγαλύτερο από 23, το υπόλοιπο της αφαίρεσης “44-Ε” δίνει την ημερομηνία της Πασχαλινής Πανσελήνου κατά το μήνα Απρίλιο ενώ αν είναι μικρότερο ή ίσο με το 23, τότε το υπόλοιπο της αφαίρεσης δίνει την ημερομηνία της Πασχαλινής Πανσελήνου το μήνα Μάρτιο του Ιουλιανού Ημερολογίου. Έτσι, στην ημερομηνία που βρέθηκε πρέπει να προστεθούν 13 ημέρες ώστε να βρεθεί η ημέρα της Πασχαλινής Πανσελήνου στο Γρηγοριανό ημερολόγιο που χρησιμοποιούμε σήμερα. Την αμέσως επόμενη Κυριακή της παραπάνω ημερομηνίας γιορτάζεται το Ελληνικό Ορθόδοξο Πάσχα.



Εικόνα 5.9: Το παράθυρο εμφάνισης της Κυριακής του Πάσχα

Στις μέρες μας έχουν αναπτυχθεί πολλοί αλγόριθμοι υπολογισμού παρέχοντας μεγαλύτερη ακρίβεια στο παραπάνω πρόβλημα. Ένας από του πιο γνωστούς αλγόριθμους είναι αυτός του μαθηματικού Carl Friedrich Gauss και του Βρετανού αστρονόμου Jean Meeus. Ο αλγόριθμος που χρησιμοποιείται στην εφαρμογή Giortes φαίνεται στη συνάρτηση easter() και είναι ο ακόλουθος :

```
void easter(int etos) {
easter_month = 4;
easter_day = ( ( 19 * (etos % 19 ) + 16 ) % 30 ) + ( ( 2 * ( etos % 4 ) ) +
                ( 4 * ( etos % 7 ) ) + ( 6 * ( ( 19 * ( etos % 19 ) + 16 ) % 30 ) ) ) % 7 + 3 ;
   if (easter_day>monthDays(easter_month, etos)) {
      easter_day -= monthDays(easter_month, etos);
      easter_month++;
   }
}
```

5.9. Εργαλεία ανάλυσης και μορφοποίησης του πηγαίου κώδικα

Κάποιοι προγραμματιστές τείνουν να έχουν το δικό τους τρόπο συγγραφής του πηγαίου κώδικα. Αυτό σημαίνει ότι κάποιος μπορεί να χρησιμοποιεί απλά κενά για τη στοίχιση του κώδικα ενώ κάποιος άλλος το πλήκτρο tab. Ακόμη, κάποιος να τοποθετεί το σύμβολο της αγκύλης στην ίδια γραμμή με το όνομα της συνάρτησης ενώ κάποιος άλλος στην επόμενη γραμμή. Παρόμοια μπορούν να υπάρχουν πολλές τέτοιες διαφοροποιήσεις. Αν και είναι σωστοί και οι δύο τρόποι συγγραφής, όταν αυτοί οι δύο προγραμματιστές συνεργάζονται πάνω στο ίδιο έργο τότε ο κώδικας του έργου γίνεται δυσανάγνωστος.

Το παραπάνω πρόβλημα λύνεται εύκολα με τη χρήση κάποιου εργαλείου μορφοποίησης κώδικα, όπως είναι το ελεύθερο λογισμικό astyle, ή πιο ολοκληρωμένα Artistic Style. Η εντολή astyle δέχεται σαν ορίσματα το στυλ της μορφοποίησης που θα παραχθεί και τα αρχεία του πηγαίου κώδικα στα οποία θα χρησιμοποιηθεί. Το astyle παρέχει αρκετές προκαθορισμένες μορφοποιήσεις, όπως το k&r, το οποίο προέρχεται από τους Kernighan & Ritchie, το gnu που χρησιμοποιείται στα περισσότερα προγράμματα του έργου GNU, το linux και άλλα. Ακόμη, παρέχει τη δυνατότητα για δημιουργία νέων στυλ όπως τα ορίσει ο χρήστης. Τα αρχεία που θα παραχθούν έχουν το ίδιο όνομα με τα αρχικά, ενώ στα αρχικά θα προστεθεί η κατάληξη .orig. Στην εφαρμογή Giortes η μορφοποίηση που χρησιμοποιήθηκε είναι η gnu.

Ένα πρόγραμμα εκτός από εύχρηστο, πρακτικό και με ευανάγνωστο κώδικα, όταν πρόκειται για ελεύθερο λογισμικό, πρέπει να είναι και ασφαλές. Γενικά είναι μια καλή πρακτική ο πηγαίος κώδικας μιας εφαρμογής να ελέγχεται για διάφορα προβλήματα ασφαλείας από κάποια εργαλεία πέρα από τον ίδιο τον προγραμματιστή της εφαρμογής. Ένα τέτοιο εργαλείο ελεύθερου λογισμικού είναι το flawfinder.

Το Flawfinder αναζητά για πιθανά κενά ασφαλείας στον πηγαίο κώδικα εφαρμογών γραμμένων στη γλώσσα προγραμματισμού C ή C++. Για τη σωστή εκτέλεσή του, το μόνο που χρειάζεται είναι να του δοθεί σαν όρισμα ο κατάλογος με τον πηγαίο κώδικα και στη συνέχεια θα γίνει αναζήτηση μέσα σε αυτόν τον κατάλογο για αρχεία τύπου C ή C++. Το Flawfinder θα παράγει μια λίστα από πιθανά κενά ασφαλείας ταξινομημένα κατά σειρά επικινδυνότητας. Το επίπεδο επικινδυνότητας εμφανίζεται μέσα σε αγκύλες και παίρνει τιμές από το μηδέν έως το 5, με το 5 να συμβολίζει τη μεγαλύτερη επικινδυνότητα. Το επίπεδο επικινδυνότητας δεν εξαρτάται μόνο από τη συνάρτηση που χρησιμοποιείται αλλά και από τα ορίσματα τα οποία παίρνει. Από προεπιλογής εμφανίζονται τα κενά ασφαλείας από το επίπεδο 1 και πάνω, ωστόσο αυτό μπορεί να αλλάξει με χρήση του ορίσματος --minlevel.

Κάθε στοιχείο της λίστας που παράγει το flawfinder δε σημαίνει απαραίτητα ότι είναι και κενό ασφαλείας, ωστόσο είναι ένα πολύ αξιόπιστο εργαλείο. Εφόσον έχει ελεγχθεί κάποιο στοιχείο της λίστας και δεν αποτελεί κενό ασφαλείας, μπορεί να επισημανθεί ο πηγαίο κώδικας ώστε να μη λαμβάνεται υπόψιν. Η επισήμανση γίνεται προσθέτοντας σαν σχόλιο το κείμενο “Flawfinder: ignore”.

Στην εφαρμογή Giortes η λίστα που παράγεται από το flawfinder είναι :

“Hits@level = [0] 0 [1] 0 [2] 10 [3] 2 [4] 0 [5] 0

Αυτό σημαίνει ότι πιθανώς να υπάρχουν δέκα κενά ασφαλείας επιπέδου 2 και δύο κενά ασφαλείας επιπέδου 3. Ωστόσο, ο κώδικας της εφαρμογής έχει ελεγχθεί εξονυχιστικά και οι παραπάνω ενδείξεις δεν αποτελούν ουσιαστική απειλή.

Αναφορές

http://library.gnome.org/devel/gtk/

http://en.wikipedia.org/wiki/Easter#Date_of_Easter

http://en.wikipedia.org/wiki/Computus

http://www.delorie.com/gnu/docs/glibc/libc.html

C Programming Language, Brian W. Kernighan, Dennis M. Ritchie, Prentice Hall PTR; 2 edition (March 31, 1988), ASIN: B001TI02S2

Beginning Linux Programming 3rd Edition. Neil Matthew, Richard Stones, Alan Cox, Wrox; 3 edition (January 2, 2004), ISBN: 978-0-7645-4497-2

The Art and Science of C: A Library Based Introduction to Computer Science, Eric S. Roberts, Addison Wesley (September 10, 1994), ISBN-13: 978-0201543223

Συμπεράσματα

Στην παρούσα πτυχιακή εργασία παρουσιάστηκε η ιστορία και η φιλοσοφία που υπάρχει πίσω από το ελεύθερο λογισμικό, καθώς και τα προτερήματα αυτού του νέου εναλλακτικού μοντέλου ανάπτυξης και χρήσης λογισμικού, που βασίζεται στην ελεύθερη διάθεση του πηγαίου κώδικα. Ακόμη, αναπτύχθηκε η Γενική Άδεια Δημόσιας Χρήσης GNU που χρησιμοποιείται στα περισσότερα έργα ελεύθερου λογισμικού και ο ανοιχτός τρόπος εξέλιξής της από την κοινότητα, φτάνοντας πλέον στην τρίτη έκδοσή της.

Παρουσιάστηκε η χρησιμότητα των εργαλειοθηκών για τη δημιουργία γραφικής διεπαφής και οι βασικότερες εργαλειοθήκες που χρησιμοποιούνται στις μέρες μας. Μεγάλη έμφαση δόθηκε στην εργαλειοθήκη GTK+ και την ανατομία των εφαρμογών που βασίζονται σε αυτή, καθώς αποτελεί μια από τις πιο αξιόλογες cross-platform λύσεις.

Αναλύθηκε η λειτουργία του GNU Build System και των εργαλείων που το αποτελούν, καθώς είναι ο κύριος τρόπος ανάπτυξης ενός έργου ελεύθερου λογισμικού στις μέρες μας. Παρουσιάστηκε ο GNU Compiler Collection, μια συλλογή μεταγλωττιστών που βοηθά στη μεταγλώττιση του έργου, και το εργαλείο Autoconf το οποίο είναι υπεύθυνο για την προετοιμασία του έργου πριν αυτό εγκατασταθεί στο σύστημα. Ακόμη, παρουσιάστηκε το εργαλείο Automake, το οποίο αναλαμβάνει τη δημιουργία κανόνων για τη σωστή μεταγλώττιση και εγκατάσταση του έργου, καθώς και το εργαλείο gettext που βοηθά στη διεθνοποίησή του.

Τέλος, παρουσιάστηκε η διαδικασία δημιουργίας ενός έτοιμου προς εγκατάσταση πακέτου, εύκολου στο διαμοιρασμό. Πιο συγκεκριμένα αναπτύχθηκε ένας τρόπος μεταγλώττισης προγραμμάτων γραμμένων για περιβάλλον Linux σε περιβάλλον Microsoft Windows, κάνοντας χρήση των εργαλείων MinGW και MSYS. Ακόμη, αναλύθηκε το πολύ ισχυρό εργαλείο NSIS το οποίο βοηθά στη δημιουργία ενός installer που αναλαμβάνει τη σωστή εγκατάσταση του έργου σε λειτουργικά συστήματα Windows. Για το λειτουργικό σύστημα Linux δημιουργήθηκαν πακέτα για τις διανομές Debian και Gentoo και παρουσιάστηκαν οι προϋποθέσεις που πρέπει να πληρεί το πακέτο ώστε να γίνουν εύκολα προσβάσιμα από τους χρήστες τους.

Η ενασχόληση με όλα τα παραπάνω με βοήθησε να κατανοήσω τον τρόπο ανάπτυξης ενός έργου ελεύθερου λογισμικού καθώς και τη λειτουργία των εργαλείων που συνήθως αυτό χρησιμοποιεί. Πλέον είμαι σε θέση να συμμετέχω ενεργά στην κοινότητα μεγάλων έργων ελεύθερου λογισμικού προσφέροντας τα μέγιστα για την ανάπτυξή τους.