ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Η χρήση του εργαλείου PyEmu στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Ταμπάκη Ειρήνη- Μαρία

Φεβρουάριος 2014

Περιεχόμενα

1. Περίληψη

Σκοπός του πονήματος αυτού είναι να παρουσιάσει όσο το δυνατόν πιο αναλυτικά η γλώσσα Python, μια γλώσσα προγραμματισμού απλή κι εύληπτη και ταυτοχρόνως αποδοτική στον αντικειμενοστραφή προγραμματισμό.

Επιπλέον, επιχειρείται προσέγγιση στη λειτουργία και τη χρήση ενός πολύ ισχυρού εργαλείου, του εξομοιωτή PyEmu.

Τέλος, γίνεται απόπειρα να αναπτυχθούν τρόποι ώστε να εφαρμοστούν τα παραπάνω σε μαθήματα της σχολής ΤΜΠΤ.

2. Εισαγωγή

Το Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών[1], έχει ως έδρα την Κοζάνη και ανήκει στην Πολυτεχνική Σχολή του Πανεπιστημίου Δυτικής Μακεδονίας. Απονέμει τίτλους σπουδών μέχρι και διδακτορικού διπλώματος και έχει ως αποστολή να παρέχει εξειδικευμένες γνώσεις σε σύγχρονους τομείς της Πληροφορικής και των Τηλεπικοινωνιών, όπως ενδεικτικά είναι η ανάλυση δεδομένων, το διαδίκτυο, η ανάλυση σήματος και εικόνας, η τεχνολογία λογισμικού και οι κινητές και δορυφορικές επικοινωνίες. Επιπλέον, στόχος του είναι να παρέχει στους φοιτητές τα απαραίτητα εφόδια που εξασφαλίζουν την άρτια κατάρτισή τους για επιστημονική και επαγγελματική σταδιοδρομία, σε εταιρίες πληροφορικής τηλεπικοινωνιών, στο δημόσιο τομέα, κ.α.

Όντας ένα πολυτεχνικό τμήμα το ΤΜΠΤ διαθέτει πληθώρα εργαστηρίων εξοπλισμένων τόσο ως προς το υλικό τους όσο και ως προς το λογισμικό. Όσον αφορά στα διαθέσιμα εργαλεία λογισμικού ενδεικτικά αναφέρω τα Matlab, Dev-C++, Opnet, Xilinx και ArgoUML. Ωστόσο, θα έλεγε κανείς ότι σε σχέση με άλλα αντίστοιχα πανεπιστημιακά τμήματα τόσο του εσωτερικού, όσο και του εξωτερικού υστερεί. Κατά την γνώμη μου, μια από τις σημαντικότερες παραλείψεις στον τομέα του λογισμικού είναι η απουσία εξομοιωτή για την γλώσσα υψηλού επιπέδου Python[2]. Όπως για παράδειγμα είναι ο PyEmu.

3. Η γλώσσα υψηλού επιπέδου Python

Η Python είναι μια εύκολη στην εκμάθηση, ισχυρή γλώσσα προγραμματισμού. Έχει αποδοτικές δομές δεδομένων υψηλού επιπέδου και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστραφή προγραμματισμό. Η κομψή σύνταξή της και οι δυναμικοί τύποι της μαζί με τη λειτουργία της ως διερμηνευόμενης (αντί μεταγλωττιζόμενης) γλώσσας την καθιστούν ΄΄ιδανική γλώσσα΄΄ για δημιουργία σεναρίων εντολών και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς και στις περισσότερες πλατφόρμες.

Πιο αναλυτικά η Python είναι ένα παράδειγμα ελεύθερου λογισμικού και λογισμικού ανοικτού κώδικα[3]. Με τον όρο λογισμικό ανοικτού κώδικα νοείται ένα λογισμικό για το οποίο δίνεται η δυνατότητα στον χρήστη να διαβάζει, να επεξεργάζεται, να διατηρεί αντίγραφα και να αξιοποιεί τμήματα του κώδικα ή ακόμα και ολόκληρο τον κώδικα παράλληλα με άλλους χρήστες. Εξαιτίας αυτού του γεγονότος, εύκολα συμπεραίνεται ότι η Python έχει τη δυνατότητα υλοποίησης σε πολλές πλατφόρμες, δεδομένου ότι όλα τα Python προγράμματά είναι δυνατό να δουλέψουν σε οποιαδήποτε από αυτές τις πλατφόρμες, χωρίς να χρειάζονται αλλαγές, με την προϋπόθεση ότι αυτά θα είναι δομημένα σύμφωνα με τις γενικές αρχές της γλώσσας. Η Python μπορεί να χρησιμοποιηθεί σε λογισμικά όπως Linux, στα Windows, στο FreeBSD, σε Macintosh, στο Solaris, στο OS/2, στην Amiga, στο AROS, στο AS/400, στο BeOS, στο OS/390, στο z/OS, στο Palm OS, στο QNX, στο VMS, στο Psion, στο Acorn RISC OS κ.α..

Προηγουμένως, η Python αναφέρθηκε ως διερμηνευόμενη αντί για μεταγλωττιζόμενη γλώσσα. Ένα πρόγραμμα που γράφεται σε μια μεταγλωττιζόμενη γλώσσα, όπως για παράδειγμα σε C ή C++, μετατρέπεται από την πηγαία γλώσσα σε μια γλώσσα που παράγεται από τον μεταγλωττιστή και λέγεται αντικείμενη γλώσσα. Η διαδικασία που ακολουθείται είναι η εξής: Ο μεταγλωττιστής δέχεται ως είσοδο ένα πρόγραμμα σε γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής, δηλαδή σε 0 και 1. Έπειτα, χρησιμοποιώντας ένα συνδέτη - φορτωτή, επιτυγχάνεται η σύνδεση του αντικείμενου προγράμματος με τις βιβλιοθήκες, ούτως ώστε να παραχθεί το εκτελέσιμο πρόγραμμα.

Η Python, ωστόσο, δε χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Αντίθετα, ο διερμηνευτής διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος, τις μετατρέπει σε μια ενδιάμεση μορφή που ονομάζεται bytecode και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία δυαδικών εντολών. Εκ των πραγμάτων η ιδιότητα αυτή της Python, κάνει τη χρήση της Python πολύ πιο εύκολη, διότι δεν απαιτούνται κατάλληλες βιβλιοθήκες, προϋποθέσεις συστήματος κλπ.. Αυτό, επίσης, κάνει τα προγράμματα σε Python εξαιρετικά φορητά.

Η Python υποστηρίζει τόσο διαδικασιοστρεφή προγραμματισμό όσο και τον αντικειμενοστρεφή προγραμματισμό. Στο διαδικασιοστρεφή προγραμματισμό, το πρόγραμμα δομείται πάνω σε διαδικασίες ή συναρτήσεις οι οποίες δεν είναι τίποτε άλλο από επαναχρησιμοποιήσιμα κομμάτια κώδικα. Στις αντικειμενοστρεφείς γλώσσες, ένα προγράμματα περιγράφει "ενέργειες" (επεξεργασίας) που εφαρμόζονται πάνω σε δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα αντικείμενα Η Python έχει έναν πολύ ισχυρό αλλά πολύ απλό τρόπο για αντικειμενοστρεφή προγραμματισμό, ειδικά όταν συγκρίνεται με άλλες πολυχρησιμοποιούμενες γλώσσες όπως η C++ ή η Java.

Δύο έννοιες που συνδέονται άρρηκτα με την γλώσσα Python είναι αυτές τις ενσωμάτωσης και της επεκτασιμότητας. Αν είναι απαραίτητο ένα κρίσιμο κομμάτι κώδικα να τρέχει, να έχει τη μέγιστη αποδοτικότητα ή κομμάτι ενός αλγόριθμου που να μην είναι ανοικτού κώδικα, τότε η γλώσσα δίνει τη δυνατότητα στον χρήστη να προγραμματίζει τμήματα κώδικα σε C ή C++ και μετά να χρησιμοποιεί αυτό στο πρόγραμμά Python. Η διαδικασία αυτή ονομάζεται ενσωμάτωση κώδικα. Η ενσωμάτωση στον προγραμματισμό Python κώδικα σε C/C++ , δίνει στον χρήστη΄τη δυνατότητα δημιουργίας σεναρίων Ο διερμηνέας Python είναι εύκολο να επεκταθεί με νέες συναρτήσεις και τύπους δεδομένων που εφαρμόζονται σε C ή C + + ή και άλλες γλώσσες που προέρχονται από την C. Η Python είναι, επίσης, κατάλληλη ως γλώσσα επέκτασης για προσαρμόσιμες εφαρμογές.

Τέλος, η Python διαθέτει πρότυπη βιβλιοθήκη η οποία είναι πραγματικά τεράστια και διατίθεται για όλες τις πλατφόρμες , με την προϋπόθεση της εγκατάστασης κάποιου εξομοιωτή Python . Αυτή περιέχει πληροφορίες σχετικά με κανονικές εκφράσεις, δημιουργία τεκμηρίωσης, δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, email, XML, HTML, αρχεία WAV, κρυπτογράφηση, γραφικές διεπαφές χρήστη (GUI -graphical user interfaces) πρωτοκόλλα Διαδικτύου (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, προγραμματισμό σε CGI), τεχνολογία λογισμικού (unit testing, καταγραφή, δημιουργία προφίλ, parsing κώδικα Python), διασυνδέσεις του λειτουργικού συστήματος (κλήσεις συστήματος, τα συστήματα αρχείων, το πρωτόκολλο TCP / IP sockets).

4. Γνωριμία με το Εργαλείο PyEmu

Δεν είναι λίγοι εκείνοι οι οποίοι χρησιμοποιούν τη Python. Η όλο και αυξανόμενη χρήση της ομολογουμένως ισχυρής αυτής γλώσσας, όπως αναλύθηκε παραπάνω, δημιούργησε την ανάγκη ανάπτυξης εργαλείων με σκοπό τη μέγιστη αξιοποίηση των δυνατοτήτων της. Το IDAPython [4], είναι μέρος του

IDAPro, που ενσωματώνει τη γλώσσα προγραμματισμού Python, επιτρέποντας στα σενάρια να εκτελούνται στο πλαίσιο του IDAPro. Τα προγράμματα αυτά έχουν πρόσβαση τόσο στα API και IDC του IDA, καθώς επίσης σε όλα τα επιπρόσθετα.

4.1 Εργαλεία βιβλιοθήκης

Εκτός από την IDAPython, αναπτύχθηκαν άλλα εργαλεία Python. Αυτά που θα μας απασχολήσουν παρακάτω, καθώς είναι απαραίτητα για τη λειτουργία του PyEmu , ανήκουν στην κατηγορία εργαλείων βιβλιοθήκης και είναι τα Pydbg[5], Pefile[6], Pydasm [7].

Όταν σε ένα πρόγραμμα συμβεί σφάλμα δύσκολο μπορούμε να το αντιληφθούμε χωρίς τη χρήση του κατάλληλου εργαλείου. Ένα εργαλείο εντοπισμού σφαλμάτων, όπως ο PyDbg, είναι ένα πρόγραμμα που έχει ως στόχο να αναλύσει άλλα προγράμματα. Το κύριο ενδιαφέρον όταν χρησιμοποιείται ένα πρόγραμμα εντοπισμού σφαλμάτων είναι να αναλυθεί ο κώδικα συμπεριφοράς ή να βρεθεί ένα σφάλμα σε κάποιο άλλο πρόγραμμα. Ένα πρόγραμμα αποσφαλμάτωσης επιτρέπει σε έναν προγραμματιστή ή έναν ερευνητή να προσδιορίσει γρήγορα την αιτία του προβλήματος στον κώδικα.

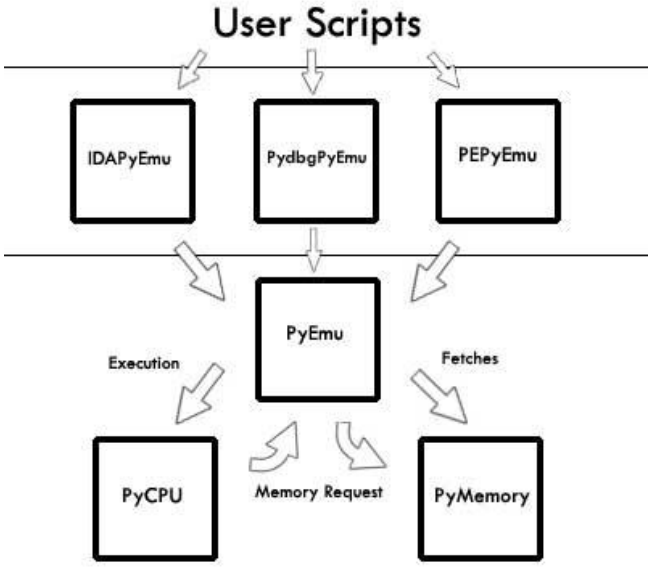
Με τη χρήση συναρτήσεων επανάκτησης που ορίζονται από το χρήστη η λειτουργικότητα της PyDbg μπορεί εύκολα να επεκταθεί. Όταν εφαρμόζεται μια ‘εθιμοτυπική’ επανάκτηση, μπορεί κανείς να ορίσει λειτουργίες παρασκηνιακά μετά την ενεργοποίηση του προγράμματος αποσφαλμάτωσης. Από την PyDbg πολλαπλές δυνατότητες είναι διαθέσιμες, όπως να ορίζονται σημεία διακοπής, να επηρεάσει ή να διαβάσει μνήμη, και να τροποποιήσει τις παραμέτρους λειτουργίας Μόλις ένα κομμάτι κώδικα εκτελεστεί μπορεί κανείς να επιστρέψει τον έλεγχο στην PyDbg, προκειμένου να συνεχιστεί η εκτέλεση.

Η Pefile είναι μια άλλη βιβλιοθήκη για το διάβασμα και την επεξεργασία εκτελέσιμων αρχείων Python τύπου PE. Αυτή η βιβλιοθήκη επιτρέπει την μεταγλώττιση των σημαντικών πληροφοριών που αφορούν σε ένα εκτελέσιμο για την αποσυναρμολόγηση συμπεριλαμβανομένων των εισαγωγών, του κώδικα και των τμημάτων δεδομένων, καθώς επίσης των διευθύνσεων των σημείων εισόδου. Η Pefile διερευνά τη διάταξη ενός αρχείου PE με σκοπό την αποτελεσματικότερη προσέγγιση σχεδόν κάθε χαρακτηριστικού του αρχείου. Ορισμένα από τα καθήκοντα της Pefile είναι η τροποποίηση και συγγραφή του αρχείου PE, η ανάλυση των ενοτήτων, η ανάκτηση δεδομένων, η προειδοποίηση για ύποπτα και δύσμορφα κέρδη. Λειτουργεί, επίσης, και ως γεννήτρια υπογραφών PEiD.

Τέλος, η pydasm, είναι ένα πρόγραμμα εξομοίωσης της βιβλιοθήκης libdasm της Python. Η Pydasm προσπαθεί να καταγράφει όλες τις οδηγίες των συναρτήσεων και αυθαίρετα να τις προσαρμόσει στον PyEmu, όπως θα δούμε παρακάτω. Το εργαλείο αυτό βοηθά τον εξομοιωτή να είναι ακόμη πιο ευέλικτος σε λειτουργία.

4.2 Το εργαλείο PyEmu

Ο εξομοιωτής PyEmu είναι ο συνδετικός κρίκος της Κεντρικής Μονάδας Επεξεργασίας της πλατφόρμας, της Κύριας Μνήμης και του χρήστη. Ανήκει στην αρχιτεκτονική x86 32-bit emulator. Από μόνος του ο PyEmu είναι ένα πολύ ισχυρό εργαλείο. Μάλιστα ακόμα και πολύ εντυπωσιακοί κώδικες ως προς το μέγεθος και τη συγγραφή, με τη χρήση του PyEmu είναι δυνατόν να εξομοιωθούν πολύ γρήγορα.



Εικόνα: Τρόπος λειτουργίας του πακέτου εξομοίωσης PyEmu

Ο PyEmu έχει ως αντικείμενο την αναγνώριση της πληροφορίας που παρέχεται από τον χρήστη, την εκτέλεση των εντολών και την ανάκτηση δεδομένων- πληροφοριών από τη μνήμη. Διαφέρει από άλλους εξομοιωτές, καθώς αποτελείται από τρία διαφορετικά, αλλά άρρηκτα συνδεδεμένα τμήματα. Αυτά είναι τα PyEmu, PyCPU και PyMemory.

Μέσο διασύνδεσης της αρχιτεκτονικής PyEmu είναι ο χρήστης, ο οποίος ελέγχει τις θέσεις μνήμης και την εκτέλεση των εντολών με σκοπό την εκτέλεση

του προγράμματος. Ο χρήστης έχει τη δυνατότητα να ελέγχει όλες τις πτυχές των διαδικασίας του εργαλείου PyEmu. Στην πραγματικότητα, ο χρήστης έχει πρόσβαση μόνο στις κλάσεις του PyEmu, ωστόσο όλες οι χρήσιμες πληροφορίες εκτίθεται μέσω δημόσιων μεθόδων κατά τη δημιουργία εμφανίσεων του PyEmu από την παραγόμενη κλάση.

Η διαδικασία που ακολουθείται, αφού ο χρήστης δώσει μια οδηγία στο υπολογιστικό σύστημα είναι η εξής. Ο PyEmu ανιχνεύει και αναγνωρίζει σε μορφή δυαδικών εντολών τις πληροφορίες μία προς μία. Ταυτόχρονο και για κάθε εντολή, δίνει οδηγίες στην PyCPU να την εκτελέσει. Επειτα, η PyCPU θα αναζητήσει τη μνήμη για την εν λόγω οδηγία και την ανάκτηση της από την PyMemory. Τα μηνύματα από την PyCPU στην PyMemory και αντιστρόφως, διαβιβάζονται μέσω του PyEmu.

Η PyCPU βασίζεται σε μια μυριάδα βοηθητικών συναρτήσεων για καθαρή ανάγνωση τιμών μητρώου και διευθύνσεων μνήμης. Χειρίζεται όλες τις λογικές εντολές, τις εντολές εκτέλεσης και των σχετικών με την εκτέλεση καθηκόντων του επεξεργαστή. Δουλειά της είναι να εκτελεί μια δεδομένη οδηγία που βασίζεται αυστηρά στις προδιαγραφές αναφοράς της Intel. Η PyCPU στηρίζεται στο μεταβλητό μέγεθος της μνήμη, ή των καταχωρητών, σε μια προσπάθεια να λειτουργεί με δημόσια τα δεδομένα της. Όπως και με κάθε κομμάτι της αρχιτεκτονικής PyEmu, ο κώδικας CPU προσπαθεί να χειριστεί αυτόνομα όλες αυτές τις απαραίτητες λειτουργίες. της

Η pyCPU αναγνωρίζει ένα πλήθος συναρτήσεων. Συνοπτικά οι συναρτήσεις αυτές εκτελούν λειτουργίες σχετικές με την ανάκτηση ενός ζητούμενου μητρώου μνήμης, ή με την ανάθεση τιμής σε αυτό, με τον υπολογισμό μιας θέσης μνήμης ή και της αξίας της. Υπάρχουν, επίσης συναρτήσεις για τον καθορισμό των κατάλληλων σημαιών CPU (λ.χ. CF, OF, SF) για μια αριθμητική πράξη και ενημέρωσης των σημαιών που χρησιμοποιούνται στην ολίσθηση(δηλ. SF, PF, και ZF). Ενδεικτικά αναφέρω τις get_register(register, size), set_memory(address, value, size) και set_shift_flags(result, size)

Η κλάση CPU δεν μπορεί να χρησιμοποιηθεί άμεσα. Ο pyEmu διεκπεραιώνει όλες τις κλήσεις για την εκτέλεση και τα αιτήματα της μνήμης.

Η PyMemory θα μπορούσε να χαρακτηριστεί ως ο χειριστής μνήμης του PyEmu, διότι οι αρμοδιότητες του περιορίζονται στην ανάκτησης από ή αποθήκευσης στις θέσεις μνήμης. Αυτή η κατηγορία είναι πολύ βασική για στην υλοποίηση. Η PyMemory βασίζεται εξ' ολοκλήρου στις εντολές που δίνονται από τον χρήστη. Δεν λειτουργεί ανεξάρτητα από τα άλλα δύο μέρη του PyEmu, άλλα παραμένει αδρανείς μέχρις ότι να κληθεί. Λειτουργεί κρατώντας μια κρυφή μνήμη των ήδη ανακτημένων σελίδων μνήμης σε τοπικό επίπεδο. Το γεγονός αυτό της δίνει τη δυνατότητα να εφαρμόζει τη διαχείριση μνήμης.

Η PyMemory διαθέτει πληθώρα μεθόδων ανάκτησης και κρυφής μνήμης. Ενδεικτικά αναφέρω τις get_page() και get_memory(). Αυτές επιτρέπουν στην PyMemory να ελέγχει εύκολα από πού προέρχονται τα δεδομένα και πως αυτά αποθηκεύονται. Για παράδειγμα, αν δεν υπάρχει στην κρυφή μνήμη μια σελίδα, θα κληθεί η συνάρτηση get_page().

Πρέπει να γίνει σαφές το γεγονός ότι PyMemory διατηρεί το τοπικό αντίγραφο ξεχωριστά από τη μνήμη πραγματικών διαδικασιών. Το εργαλείο PyEmu δεν έχει πρόσβαση στην κρυφή μνήμη της PyEmu. Αυτό συμβαίνει με σκοπό τη διασφάλιση όλων των σελίδων που χρησιμοποιούνται από τον εξομοιωτή κατά τη διαδικασία της αποσφαλμάτωσης. Παρατηρούμε, λοιπόν, ότι σε περίπτωση σφάλματος μια καλή λύση θα ήταν η απόρριψη όλων των σελίδων που βρίσκονται στην πραγματική μνήμη και η αντικατάσταση αυτών από την PyMemory.

4.2 Η χρήση του PyEmu

Η χρήση του PyEmu πρέπει να είναι εύκολη και αποτελεσματιική. Περιλαμβάνει τη δημιουργία αντικειμένων, ανάκτηση από και καταγραφή στη μνήμη και εκτέλεση. Στηρίζεται μια λογική ακολουθία συγκεκριμένων βημάτων την επίτευξη των στόχων που ο χρήστης θέτει και προσπαθεί να επιλύσει με τη χρήση του εξομοιωτή PyEmu.

Η πιο βασική δράση κατά την χρήση του PyEmu είναι η εκτέλεση μιας εντολής. Για την ανάκτηση αυτής την εντολής η διαδικασία έχει ως εξής, μόλις η η PyCPU ανακτήσει και αντιγράψει τις κατάλληλες θέσεις μνήμη του τρέχοντα δείκτη διεύθυνσης που χειρίζεται μια συγκεκριμένη εντολή και τα παραδίδει αρχικά στην βιβλιοθήκη pydasm. Το Pydasm μας επιτρέπει να αποκωδικοποιούμε σωστά την εντολή. Αυτή η απλή λειτουργία είναι η ουσία της PyEmu. Η εξυπηρέτηση διαφόρων συναρτήσεων, όπως αυτές των IDA Pro, Pydbg και Pefile η αποκωδικοποίηση βήμα- βήμα, επιτρέπει στον εξομοιωτή την αυθαίρετη αποκωδικοποίηση των οδηγιών. Ο PyEmu επιδιώκει, λοιπόν, να είναι εύκολη η επέκταση, μνήμης και κώδικα χειρισμού που έχουν προ-δημιουργηθεί και επιτρέπουν την ομοιόμορφη εμφάνιση και τη λειτουργία όλων των οδηγιών.

Η χρήση του εργαλείου του PyEmu, ωστόσο έχει και ορισμένα προβλήματα. Όπως είπαμε ο χρήστης εντάσσεται στην κλάση του PyEmu και συνεργάζεται με τις παρεχόμενες συναρτήσεις του, ακριβώς με τον τρόπο που ορίζεται από τον PyEmu. Αυτό το χαρακτηριστικό προκαλεί προβλήματα κατά τη συγγραφή του σεναρίου χρησιμοποιώντας τον PyEmu,. Για παράδειγμα, όταν εργάζεται στο IDA Pro, ένας χρήστης θα θέλει να χρησιμοποιήσετε την κλάση

IDAPyEmu καθώς παρέχει την πρόσθετη υποστήριξη που χρειάζεται κατά τη διάρκεια της εγκατάστασης. Καλό θα ήταν, λοιπόν, ο χρήστης πρώτα να δημιουργεί και τη δική τους κλάση εξομοίωσης.

5. Αξιοποίηση του PyEmu στο ΤΜΠΤ

Η εργασία αυτή δεν αποσκοπεί στην απρόσκοπτη εύρεση εφαρμογών του PyEmu στο τμήμα και ένταξης της Python στον οδηγό σπουδών. Παρακάτω, αναλύονται μερικά μόνο μαθήματα όπως αυτό της Αρχιτεκτονικής Υπολογιστών και του Δομημένου Προγραμματισμού, για τα οποία ενώ θα μπορούσε κανείς να πει πως οι εφαρμογή της Python δεν μπορεί να υπάρξει, όσον αφορά το εργαστηριακό μέρος των μαθημάτων θα διακαυθεί. Ωστόσο, όπως θα δούμε υπάρχουν περιπτώσεις, που η χρήση της δεν κρίνεται απαραίτητη.

5.1 Python, PyEmu και Προγραμματισμός

Με μια γρήγορη ματιά στον οδηγό σπουδών του Τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, θα συναντήσει κανείς μαθήματα όπως Δομημένος Προγραμματισμός, Αντικειμενοστραφής Προγραμματισμός Ι και ΙΙ, τα οποία καταπιάνονται με την εκμάθηση των γλωσσών υψηλού επιπέδου C, C++ και Java αντίστοιχα.

Όσον αφορά το μάθημα του Δομημένου Προγραμματισμού, το οποίο ανήκει στο πρώτο χειμερινό εξάμηνο, θα μπορούσε κανείς να το θεωρήσει το σημαντικότερο μάθημα προγραμματισμού του τμήματος. Είναι γεγονός, πως το υπάρχον σύστημα εισαγωγής στα Ανώτατα Εκπαιδευτικά Ιδρύματα με αντικείμενο την πληροφορική, τον προγραμματισμό και τα δίκτυα τηλεπικοινωνιών, επιτρέπει την εισαγωγή φοιτητών, χωρίς απαραίτητη προϋπόθεση την γνώση βασικών δομών προγραμματισμού. Η γλώσσα C, ως βάση άλλων γλωσσών προγραμματισμού, όπως C++ και C#, είναι ανάγκη να παρατίθεται στους φοιτητές του πρώτου έτους αυτόνομη και αναλυτική. Η παράλληλη διδασκαλία της python ή η παράθεση όλων αυτών των μέσων και των περιορισμών για τη συνεργασία Python και C, πιθανότατα να μην μπορεί να κατανοηθεί πλήρως από τους φοιτητές, με αποτέλεσμα την αποθάρρυνση τους.

Προφανέστατα, δεν ισχύει το ίδιο και για τις άλλες δύο γλώσσες όπως αναφέρθηκαν παραπάνω. Διότι τα μαθήματα, του αντικειμενοστραφή προγραμματισμού θα μπορούσαν να αντικατασταθούν ή σε ακόμα καλύτερη περίπτωση να εμπλουτισθούν με την εκμάθηση της Python και του PyEmu simulator. Εδώ, κύριο μέλημα, των καθηγητών είναι η ανάπτυξη των ικανοτήτων του φοιτητή και η δημιουργία ερεθισμάτων σχετικά με την συνεργασία και όχι μόνο των γλωσσών αυτόν. Επιπλέον, πλήθος χρηστών αναρτούν κριτικές στο διαδίκτυο εξυμνώντας την Python και τα εργαλεία της. Υποστηρίζουν ότι η γλώσσα αυτή βοήθησε στην ανάπτυξη αποδοτικότερων προγραμμάτων με ελάχιστο χρόνο εκτέλεσης. Κάπου εδώ πρέπει να τονίσουμε ότι, σκοπός της εκπαίδευσης ενός μηχανικού, δεν είναι απλά η λήψη γενικών γνώσεως, αλλά η απόλυτη γνώση και η ανάπτυξη κριτικής ικανότητας με σκοπό την ορθότερη και αποτελεσματικότερη εφαρμογή της.

5.2 Ο PyEmu στην Αρχιτεκτονική Υπολογιστών

Όπως προαναφέρθηκε η Python μέσω του διερμηνευτή, έχει την ικανότητα εκτέλεσης των προγραμμάτων της εντολή προς εντολή. Από την άλλη μεριά ο εξομοιωτής PyEmu , λειτουργεί ως συνδετικός κρίκος των βασικότερων για την αρχιτεκτονική των υπολογιστών δομών, λαμβάνοντας ως παράγοντα τον χρήστη και τα δεδομένα- πληροφορίες που αυτός εισάγει.

Αν ανατρέξει, κανείς στην ηλεκτρονική πλατφόρμα του τμήματος eclass.uowm.gr . Θα παρατηρήσει, ότι το εργαστήριο υλοποιείται σε γλώσσα Assembly στον εξομοιωτή Emu8086. Εξ' ορισμού η γλώσσα Assembly είναι μια γλώσσα προγραμματισμού χαμηλού επιπέδου για έναν υπολογιστή ή οποιαδήποτε άλλη προγραμματιζόμενη συσκευή, στην οποία υπάρχει μια πολύ ισχυρή αντιστοιχία μεταξύ της γλώσσας και των εντολών κώδικα μηχανής της αρχιτεκτονικής.

Φαίνεται, λοιπόν, σιγά-σιγά να κτίζεται αναπτύσσεται ένας συνδετικός κρίκος μεταξύ αυτών των δύο γλωσσών. Όμως, κάθε συμβολική γλώσσα είναι ειδικά σχεδιασμένη για μία συγκεκριμένη αρχιτεκτονική υπολογιστών. Η Python, όπως, άλλωστε και οι περισσότερες γλώσσες υψηλού επιπέδου είναι φορητή σε πολλά αρχιτεκτονικά συστήματα. Επιπλέον, το χαρακτηριστικό ότι είναι γλώσσα ΕΛ/ΛΑΚ, την καθιστά την καταλληλότερη για την αρχιτεκτονική υπολογιστών. Η χρήση του PyEmu μας, εξασφαλίζει την δυνατότητα στον χρήστη παρακολούθησης , τόσο των καταχωρητών της μνήμης όσο και την εκτέλεση των οδηγιών και την αποτελμάτωση.

Τέλος, θεωρώ απαραίτητο να αναφέρω, πως δεν είναι λύση η πλήρης αντικατάσταση των Assembly και Emu8086 με τα αντίστοιχα Python και PyEmu. Ακόμα και κάποιος αντιτάξει, ότι με την Python θα υλοποιούνται προγράμματα μικρότερα και πολύ γρήγορα στην εκτέλεση, μια τέτοια πράξη να είναι καταστροφική, για τους μελλοντικούς μηχανικούς. Δεν είναι δυνατόν, να αγνοούν ένα κομμάτι τόσο σημαντικό του προγραμματισμού. Όμως, η παράλληλη επεξεργασία των γλωσσών αυτών, με την υλοποίηση, εργαστηριακών ασκήσεων τόσο για τον Emu8086 όσο και για τον PyEmu, θα βοηθήσει στην αποσαφήνιση της διαδικασίας μετάβασης από γλώσσες υψηλού επιπέδου σε γλώσσα μηχανής και αντίστροφα, μέσω των γλωσσών χαμηλού επιπέδου.

6. Συμπεράσματα

Ολοκληρώνοντας αυτή τη μικρή μελέτη, όλα δείχνουν ότι η Python είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου που με την εξομοίωσή της στο εργαλείο PyEmu, είναι εφαρμόσιμη σε τμήματα της πληροφορικής, τα οποία σχετίζονται περισσότερο με το υλικό, παρά με το λογισμικό. Την δυνατότητα αυτή, την έχουν ελάχιστες γλώσσες προγραμματισμού, ενώ ακόμη λιγότερα εργαλεία όπως αυτό του PyEmu έχουν δημιουργηθεί.

Δεδομένου του αυξημένου ανταγωνισμού στον εργασιακό τομέα και της ανάγκης για εξειδίκευση, η Python σε συνάρτηση με το εργαλείο PyEmu είναι ένα ισχυρό προσόν για την μελλοντική επιτυχία κάθε μηχανικού. Είναι μία νέα γλώσσα προγραμματισμού που συνεχώς εξελίσσεται και προσαρμόζεται στα νέα δεδομένα. Θεωρώ, ότι η γνώση των παραπάνω είναι υποχρεωτική, για κάθε φοιτητή του τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών.

7.Αναφορές

[1] Πανεπιστήμιο Δυτικής Μακεδονίας – Πολυτεχνική Σχολή
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
http://icte.uowm.gr/index.php?page=1

[2] Η επίσημη σελίδα της Python
http://www.python.org/

[3] Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα (ΕΛ/ΛΑΚ)
http://www.ellak.gr/

[4] PyEmu: A multi-purpose scriptable IA-32 emulator
https://www.blackhat.com/presentations/bh-usa-07/Pierce/Whitepaper/bh-usa- 07-pierce-WP.pdf

[5]IDAPython
http://d-dome.net/idapython/

[6]Pydbg
http://py.french.taggy.cz/pydbg-tutorial/

[7]Pefile
https://code.google.com/p/pefile/

[8]Pydasm
http://libdasm.googlecode.com/svn/trunk/pydasm/README.txt