

Προηγμένοι Μικροεπεξεργαστές

Παρουσίαση Projects

Γενικά

- 5 Projects για ένα άτομο
- 6 Projects για δύο άτομα
- 1 Project για ένα έως δύο άτομα
- 3 Projects για δύο έως τρία άτομα

Γενικά

- Προθεσμία:
 - Τέλος εξεταστικής (Hard Deadline)
- Τυπικός απαιτούμενος χρόνος:
 - 2-3 μέρες προετοιμασία
 - 3-8 μέρες δουλειά full time
- Παραδοτέα:
 - Κώδικας και αναλυτική αναφορά

Debugging

- Το debugging είναι επίσης κομμάτι της δουλειάς σας
- Πριν ζητήσετε βοήθεια:
 - Ψάξτε στο manual της Intel και στο δίκτυο
 - Περιορίστε το που μπορεί να οφείλεται το πρόβλημα

Κώδικας

- Ο κώδικας πρέπει να είναι:
 - Οργανωμένος (συναρτήσεις, structs, macros, ξεχωριστές λειτουργικότητες σε ξεχωριστά αρχεία)
 - Καθαρογραμμένος (περιγραφή συναρτήσεων, σχόλια μέσα στον κώδικα, ονόματα που βοηθούν την κατανόηση, στοίχιση του κώδικα)

Θέμα 1^ο – Wrapper για RM interrupts από PM

- Υλοποίηση ISR για τα interrupts 0x14 και 0x21
- Θα χρησιμοποιούν Virtual Mode για να καλέσουν τα αντίστοιχα Real Mode Interrupts
- Για 2 άτομα

Θέμα 1^ο – Wrapper για RM interrupts από PM

- Το virtual mode είναι ένα mode συμβατότητας
 - Ναι μεν protected mode, αλλά με real mode address translation → μπορούμε να εκτελέσουμε rm κώδικα
- Περνάμε σε virtual mode, αν τεθεί το VM flag στον eflags
- Επιστρέφουμε σε protected mode, αν γίνει reset το VM flag ή εκτελεστεί sensitive εντολή

Θέμα 1^ο – Wrapper για RM interrupts από PM

- Οι ISR μας θα καλούν ένα virtual mode task
 - Το vm flag στο TSS του πρέπει να είναι 1
 - Οι segment registers στο TSS αρχικοποιημένοι με rm τιμές
 - Ο EIP θα δείχνει στον interrupt handler που θέλουμε να εκτελέσουμε
 - Οι υπόλοιποι καταχωρητές στο TSS πρέπει να έχουν τις τιμές με τις οποίες θέλουμε να καλέσουμε το service.
- Ο επεξεργαστής θεωρεί τα vm tasks ότι έχουν privilege level 3
 - Το πεδίο IOPL στον eflags πρέπει να είναι 3

Θέμα 1^ο – Wrapper για RM interrupts από PM

- References:
 - 386 Intel Manual: Chapter 15

Θέμα 2^ο – Υλοποίηση loader os

- Δυναμική φόρτωση και εκτέλεση binaries που έχουν υλοποιηθεί ανεξάρτητα από τον υπόλοιπο κωδικά μας
- Για 2-3 άτομα

Θέμα 2^ο – Υλοποίηση loader os

- Ο κώδικας μας πρέπει
 - να διαβάζει το αρχείο που περιέχει το binary
 - να το αντιγράφει σε μία ελεύθερη περιοχή της μνήμης
 - να προσθέτει εγγραφές για τους code, data και tss descriptors στον GDT
 - να προσθέτει ένα νέο TSS
 - να κάνει task switch στο νέο task

Θέμα 2^ο – Υλοποίηση loader os

- Για να λειτουργήσει αυτή η διαδικασία πρέπει:
 - Ο κώδικας του binary να μην αλλάζει τους segment registers
 - Να δώσουμε CS και DS στο νέο task, ώστε να καλύπτουν την περιοχή όπου φορτώσαμε το πρόγραμμα.

Θέμα 3^ο – Υλοποίηση FIR φίλτρου

- Δύο υλοποιήσεις FIR φίλτρου:
 - Μία με x87 εντολές
 - Μία με SSE εντολές
- Σύγκριση των δύο υλοποιήσεων
- Για δύο άτομα

Θέμα 3^ο – Υλοποίηση FIR φίλτρου

- Οι x87 εντολές χρησιμοποιούν τον μαθηματικό συνεπεξεργαστή.
 - Δύστροπος και αργός
- Οι sse εντολές προστίθενται από τους pentium III και μετά
 - Πιο εύκολες, πιο γρήγορες και με δυνατότητες για vector πράξεις

Θέμα 3^ο – Υλοποίηση FIR φίλτρου

- Και οι δύο υλοποιήσεις θα γραφτούν σε κανονική C, με assembly συναρτήσεις για την χρήση του συνεπεξεργαστή και των sse εντολών
- Το φίλτρο
 - θα λειτουργεί πάνω σε float (32-bit)
 - θα χρησιμοποιεί τα τελευταία 8 δείγματα
 - στην sse υλοποίηση πρέπει να κάνει vector πράξεις σε packed δεδομένα
- Τα δείγματα θα περιέχονται σε έναν πίνακα 1024 θέσεων, τον οποίο θα ανακυκλώνουμε 100 φορές

Θέμα 3^ο – Υλοποίηση FIR φίλτρου

- Στο τέλος θα κάνετε μία απλή σύγκριση της ταχύτητας των δύο υλοποιήσεων:
 - Πχ σε linux με την εντολή `time`
- Για να είναι στατιστικά έγκυρες οι μετρήσεις, πρέπει να πάρετε τον μέσο όρο από πολλαπλές εκτελέσεις και να τρέχετε όσο το δυνατόν λιγότερα προγράμματα ταυτόχρονα.
- References:
 - Καινούργια manuals: Section 5.2, 5.5 – 5.7
 - http://en.wikipedia.org/wiki/Streaming_SIMD_Extensions
 - <http://en.wikipedia.org/wiki/X87>

Θέμα 4^ο – Απεικόνιση MBR

- Assembly πρόγραμμα που θα διαβάσει τον δίσκο με bios services και θα εκτυπώνει τα partitions και όλες τις πληροφορίες για αυτά που περιγράφονται στο mbr
- Για ένα άτομο
- References:
 - http://en.wikipedia.org/wiki/Master_boot_record

Θέμα 5^ο – Paging με swapping

- Πάνω σε υπάρχοντα κώδικα για paging να υλοποιηθεί swapping
- Για ένα άτομο

Θέμα 5^ο – Paging με swapping

- Swap out:
 - Ανά 5 δευτερόλεπτα, ο κώδικας θα διαβάζει το page table για να βρει εγγραφές που δεν έχουν χρησιμοποιηθεί
 - Θα αντιγράψει την αντίστοιχη περιοχή της μνήμης σε ένα αρχείο στον δίσκο (swarfile)
 - Θα διαγράψει την εγγραφή από το page table
- Swap in:
 - Αν χτυπήσει Page Fault, ο κώδικας θα ψάξει στο swarfile για το page που ζητήθηκε και αν το βρει θα το μεταφέρει στην μνήμη

Θέμα 5^ο – Paging με swapping

- Ένα τογ προγραμμα θα διαβάζει ψευδοτυχαίες θέσεις μνήμης μέσα από έναν πίνακα 32KB, με ρυθμό περίπου 2 accesses / second, ώστε να ελέγξουμε την λειτουργία του swapping.
- References:
 - 386 intel manual: Section 5.2 + Section 9.8.14

Θέμα 6^ο – Απλός debugger

- Υλοποίηση ενός απλού debugger χρησιμοποιώντας άμεσα τα debug facilities των x86.
- Για 2-3 άτομα

Θέμα 6^ο – Απλός debugger

- Ο debugger θα υλοποιηθεί σαν καινούργιο task και θα πρέπει να έχει τις εξής interactive δυνατότητες:
 - “Συγκόληση” πάνω σε υπάρχον task
 - Δημιουργία breakpoint μέσω των debug registers
 - Εκτύπωση των περιεχομένων όλων των registers
 - Εκτύπωση της τιμής οποιασδήποτε θέσης μνήμης
 - Συνέχιση της εκτέλεσης του προγράμματος

Θέμα 6^ο – Απλός debugger

- References:
 - Νέα Intel manual: Chapter 18

Θέμα 7^ο – Bmp Display

- Υλοποίηση ενός C προγράμματος πάνω στον κώδικα του εργαστηρίου που θα απεικονίζει στην οθόνη ένα bmp αρχείο
- Για 1-2 άτομα

Θέμα 7^ο – Bmp Display

- Χρήση της VGA σε graphics mode
 - 640x480 16 colors
- Φόρτωση μέσω DOS services της εικόνας από αρχείο στην μνήμη
- Μετατροπή της εικόνας στο format της VGA
- Εγγραφή στην video memory
 - Graphics mode: 0xA0000 (64 KB)

Θέμα 8^ο – File System

- Υλοποίηση σε C πάνω στον κώδικα του εργαστηρίου ενός στοιχειώδους File System, χρησιμοποιώντας BIOS Services για την προσπέλαση της δισκέτας.
- Για 2-3 άτομα

Θέμα 8^ο – File System

- Θα παρέχει υπηρεσίες
 - ανάγνωσης αρχείου από την δισκέτα σε memory buffer
 - εγγραφής από memory buffer σε νέο αρχείο
 - διαγραφή αρχείου
 - δημιουργία αρχείου
 - Μία απλή γραμμή εντολών θα παρέχει τρεις εντολές:
 - Store: θα διαβάζει το πληκτρολόγιο και θα γράφει σε νέο αρχείο
 - Print: θα εμφανίζει το αρχείο στην οθόνη
 - Format: θα δημιουργεί το file system στην δισκέτα.

Θέμα 8^ο – File System

- Υποτυπώδες file system:
 - Τα ονόματα των αρχείων είναι μέχρι οχτώ χαρακτήρες
 - Μέχρι 16 Αρχεία
 - Τα αρχεία είναι συνεχόμενα στο δίσκο
 - Ανάμεσα στα αρχεία δεν υπάρχουν κενά
 - Όλος ο ελεύθερος χώρος βρίσκεται μετά το τέλος του τελευταίου αρχείου
 - Ένα Allocation Table στο πρώτο sector περιέχει ονόματα αρχείων, block από όπου αρχίζουν και block όπου τελειώνουν

Θέμα 9^ο – Stack Protection

- Επέκταση του κώδικα του εργαστηρίου ώστε να υλοποιηθεί ένα μηχανισμός stack protection μέσω του limit protection
- Για ένα άτομο

Θέμα 9^ο – Stack Protection

- Ο μηχανισμός θα χρησιμοποιεί το stack exception για να αναγνωρίζει προσπελάσεις στοίβας που ξεφεύγουν από τον χώρο που έχουμε αναθέσει στην στοίβα
- Όταν συμβεί αυτό, η στοίβα θα αντιγράφεται σε νέο segment και θα ανανεώνεται ο ss του προγράμματος με το νέο segment

Θέμα 9^ο – Stack Protection

- Για να το έλεγξουμε, θα γραφτεί ένα μικρό πρόγραμμα σε privilege level 3:
 - Θα το δίνουμε stack όπως μέχρι τώρα (σχετικά μικρό), αλλά το stack segment του θα έχει limit που να καλύπτει μέχρι την κορυφή της στοίβας.
 - Το πρόγραμμα θα κάνει recursive κλήσεις σε μία μικρή συνάρτηση, ώστε να γεμίσει την στοίβα του

Θέμα 9^ο – Stack Protection

- References:
 - 386 Intel manual: Section 6.3.1.2 και 9.8.12

Θέμα 10^ο – Virtual Machine Extensions

- Μελέτη και παρουσίαση των Virtual Machine Extensions
 - Πως στήνεται βήμα προς βήμα ένα VMX περιβάλλον
 - Πως ο hypervisor ελέγχει και παρεμβαίνει στην εκτέλεση μιας διεργασίας.
- Για ένα άτομο

Θέμα 11^ο – Παραγωγός-Καταναλωτής

- Υλοποίηση παραδείγματος παραγωγού-καταναλωτή
 - Η επικοινωνία μεταξύ των δύο tasks να γίνεται με κοινόχρηστο buffer και σημαφόρο
 - Αξιοποίηση των ατομικών εντολών της x86 αρχιτεκτονικής.
- Για ένα άτομο

Θέμα 12^ο – PC Speaker

- Υλοποίηση οδηγών για το PC Speaker
 - Η διεπαφή των οδηγων θα δέχεται συχνότητα και διάρκεια ήχου
 - Ένα μικρό προγράμμα να αναπαράγει ένα μικρό κομμάτι μουσικής με χρήση αυτών των οδηγών
- Για δύο άτομα
- Reference:
 - <http://fly.cc.fer.hr/GDM/articles/sndmus/speaker1.html>

Θέμα 13^ο – DVFS

- Υλοποίηση προγράμματος που θα παρέχει ένα απλό τρόπο για αλλαγή της συχνότητας λειτουργίας του επεξεργαστή
- Αποκλειστικά σε linux με Intel επεξεργαστή
- Για 1 άτομο

Θέμα 13^ο – DVFS

- Οι intel επεξεργαστές προσφέρουν ένα σύστημα για γρήγορες αλλαγές συχνότητας και τάσης λειτουργίας
 - Enhanced Intel Speedstep
- Ο πυρήνας του linux αξιοποιεί τις dvfs δυνατότητες των intel επεξεργαστών μέσω του module cpufreq
 - Παρέχει μία σειρά από in-kernel governors (πολιτικές dvfs) καθώς και userspace διεπαφές μέσω του /sys

Θέμα 13^ο – DVFS

- Στόχος του project είναι να γραφτεί ένα πρόγραμμα σε linux που:
 - Θα παρακολουθεί το πληκτρολόγιο
 - Αν πατηθεί το F12 θα ανεβάζει το πάνω και κάτω όριο των επιτρεπτών συχνοτήτων λειτουργίας μέσω της /sys διεπαφής
 - Αν πατηθεί το F11, θα κατεβάζει τα όρια των συχνοτήτων λειτουργίας
- References:
 - <http://software.intel.com/en-us/articles/enhanced-intel-speedstep-technology>

Θέμα 14^ο – PerfCounters 1

- Χρήση Performance Counters για μελέτη εφαρμογής κωδικοποίησης video, όταν αυτή γίνεται compile με διαφορετικές εκδόσεις του gcc και διαφορετικά optimization flags
- Για 2 άτομα

Θέμα 14^ο – PerfCounters 1

- Προτείνεται:
- Αλγόριθμος κωδικοποίησης → xvid
- Compilers → gcc4.1, gcc4.3 (gcc3.4)
- Compiler flags → march, -O0/O1/O2/O3, -use-sse3

Θέμα 15^ο – PerfCounters 2

- Διερεύνηση με performance counters της συμπεριφοράς διαφόρων java virtual machines για το ίδιο εκτελούμενο πρόγραμμα
- Αποκλειστικά για linux
- Για δύο άτομα

Θέμα 15^ο – PerfCounters 2

- Η Java virtual machine:
 - μετάφραζει τα java bytecodes
 - στήνει το java περιβάλλον

Θέμα 15^ο – PerfCounters 2

- Εκτός από την υλοποίηση της Sun (Hot Spot JVM) υπάρχει μία σειρά από άλλες virtual machines
 - Jikes RVM
 - Low Level Virtual Machine (LLVM)
 - Kaffe
 - JamVM
- διαφορετικές λογικές λειτουργίας
- πειραματικές ιδιότητες και βελτιστοποιήσεις
- έμφαση σε διαφορετικά είδη προγραμμάτων χρήστη

Θέμα 15^ο – PerfCounters 2

- Για να μελετήσετε τις VM, χρησιμοποιήστε τα παρακάτω benchmarks:
 - <http://www.dacapobench.org/>
 - <http://www.netlib.org/benchmark/linpackjava/>

Θέμα 16^ο – PerfCounters 3

- Διαγωνισμός για την υλοποίηση σε πραγματικό σύστημα, αλγορίθμου Gaussian blur.
- 1 άτομο vs 1 άτομο

Θέμα 16^ο – PerfCounters 3

- Το πρόγραμμα θα διαβάζει ένα pgm ascii αρχείο και θα εκτελεί Gaussian blur με βάση το matrix που θα βρείτε εδώ:
 - http://en.wikipedia.org/wiki/Gaussian_blur#Sample_Gaussian_matrix
- Ο αλγόριθμος να γραφτεί σε C, με δυνατότητα για υλοποίηση τμημάτων σε assembly.
- Τα προγράμματα θα αξιολογηθούν για optimization levels 0 και 3

PerfCounter γενικά

- Τα metrics που μας ενδιαφέρουν είναι:
 - Execution time
 - L1 accesses και misses
 - L2 accesses και misses
 - Committed branches και branch mispredictions
- Για την χρήση των performance counters προτείνεται το Vtune της Intel
 - <http://software.intel.com/en-us/articles/intel-software-evaluation-center/>
 - Το license είναι για 30 μέρες, οπότε εγκαταστήστε το αφού έχετε τελειώσει με τα υπόλοιπα κομμάτια των projects