

ΠΕΡΙΓΡΑΜΜΑ ΜΑΘΗΜΑΤΟΣ

(1) ΓΕΝΙΚΑ

ΣΧΟΛΗ	Μηχανικών		
ΤΜΗΜΑ	Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών		
ΕΠΙΠΕΔΟ ΣΠΟΥΔΩΝ	Προπτυχιακό		
ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ	9.011	ΕΞΑΜΗΝΟ ΣΠΟΥΔΩΝ	9 ^ο
ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ	Λογισμικό Συστήματος		
ΑΥΤΟΤΕΛΕΙΣ ΔΙΔΑΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ		ΕΒΔΟΜΑΔΙΑΙΕΣ ΩΡΕΣ ΔΙΔΑΣΚΑΛΙΑΣ	ΠΙΣΤΩΤΙΚΕΣ ΜΟΝΑΔΕΣ
Θεωρητικές διαλέξεις		3	2
Ασκήσεις πράξης		1	1
Εργαστηριακές ασκήσεις		1	1
ΣΥΝΟΛΟ		5	4
ΤΥΠΟΣ ΜΑΘΗΜΑΤΟΣ	Εμβάθυνσης / Εμπέδωσης γνώσεων ειδικότητας		
ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ	8.009 - Λειτουργικά Συστήματα		
ΓΛΩΣΣΑ ΔΙΔΑΣΚΑΛΙΑΣ και ΕΞΕΤΑΣΕΩΝ	Ελληνική		
ΤΟ ΜΑΘΗΜΑ ΠΡΟΣΦΕΡΕΤΑΙ ΣΕ ΦΟΙΤΗΤΕΣ ERASMUS	Όχι		
ΗΛΕΚΤΡΟΝΙΚΗ ΣΕΛΙΔΑ ΜΑΘΗΜΑΤΟΣ (URL)	https://eclass.hmu.gr/courses/ECE149		

(2) ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Μαθησιακά Αποτελέσματα
Σκοπός του μαθήματος είναι η μελέτη σε βάθος του λογισμικού συστήματος, επικεντρώνοντας στον προγραμματισμό σε επίπεδο συστήματος (systems/network programming) με βάση βιβλιοθήκες συστήματος (POSIX, SYSTEM V), όσο και σε επίπεδο kernel. Δίδεται έμφαση στην πειραματική εξάσκηση των φοιτητών σε αρκετές ενσωματωμένες πλατφόρμες (Arduino, Odroid XU4, 96boards Hikey & Dragonboard 410c, Renesas Synergy S7G2) με διαφορετικές αρχιτεκτονικές και λειτουργικά συστήματα (Linux, free/commercial RTOS) διασυνδεδεμένες με διάφορες τεχνολογίες επικοινωνιώνόπως peripheral buses, Ethernet, real-time communications. Με την επιτυχή ολοκλήρωση του μαθήματος οι φοιτητές θα είναι σε θέση: • Να μπορούν να καταλάβουν κώδικα που σχετίζεται με Linux, Real-time Operating Systems είτε σε επίπεδο IPC, είτε σε επίπεδο (top-/bottom-half) device driver ή Linux kernel module • Να γνωρίζουν την χρησιμότητα του λογισμικού συστήματος • Να έχουν κατανοήσει σε βάθος θέματα διαχείρισης πόρων, διαδιεργασιακής επικοινωνίας και drivers τόσο σε Linux όσο και σε ελεύθερα/βιομηχανικά λειτουργικά συστήματα πραγματικού χρόνου • Να αποκτήσουν ευχέρεια σε θέματα system software development, debugging, performance/energy profiling, network interfacing, small software stacks, και reverse engineering • Να μπορούν να αναπτύσσουν platform-independent GNU software χρησιμοποιώντας C/C++ & Linux και αυτόματα εργαλεία ανάπτυξης (ac-tools, autoconf, automake, κλπ) • Να αντιμετωπίζουν θέματα χρονο-δρομολόγησης διαδικασιών σε Linux • Να σχεδιάζουν και υλοποιούν Linux kernel modules και προγράμματα οδηγών για πραγματικά ή virtual devices • Να προγραμματίζουν αποδοτικά παράλληλες/κατανεμημένες εφαρμογές (π.χ. distributed embedded systems, cyber-physical systems) που χρησιμοποιούν low-level physical/virtual interfaces και βιβλιοθήκες

συστήματος • Να επεκτείνουν ανοικτό λογισμικό συστήματος
Γενικές Ικανότητες Το μάθημα αποσκοπεί στην απόκτηση, από τον πτυχιούχο, των παρακάτω γενικών ικανοτήτων: • Αναζήτηση, ανάλυση και σύνθεση δεδομένων και πληροφοριών, με τη χρήση και των απαραίτητων τεχνολογιών • Προσαρμογή σε νέες καταστάσεις • Αυτόνομη εργασία • Ομαδική εργασία • Εργασία σε διεπιστημονικό περιβάλλον • Παραγωγή νέων ερευνητικών ιδεών

(3) ΠΕΡΙΕΧΟΜΕΝΟ ΜΑΘΗΜΑΤΟΣ

Ενότητες Θεωρητικών Διαλέξεων

Αρχικά εξετάζονται θέματα ανάπτυξης ανοικτού λογισμικού και βιβλιοθηκών εξετάζοντας μεθόδους και εργαλεία ανάπτυξης λογισμικού που βασίζονται κυρίως σε GNU software development. Τα εργαλεία αυτά αφορούν κυρίως compiling, assembling, linking, και debugging σε διάφορα συστήματα, π.χ. desktop, servers, και ενσωματωμένα συστήματα.

Κεντρικό σημείο αναφοράς αποτελεί η χρήση προγραμματιστικών τεχνικών για αύξηση της απόδοσης και μείωση της ενεργειακής κατανάλωσης (dynamic power consumption). Γίνεται εκτενής αναφορά στη χρήση παράλληλων διεργασιών, νημάτων, δια-διεργασιακής επικοινωνίας (IPC). Επίσης εξετάζονται τεχνικές επιτάχυνσης σε επίπεδο compiler (SSE vector operations), μέθοδοι που αυξάνουν το locality (caching, prefetching) και εργαλεία ανάλυσης και βελτιστοποίησης (performance tuning).

Σε επίπεδο Linux kernel εξετάζεται η παραμετροποίηση (configuration, device trees και μετάφραση του πυρήνα (cross compiling toolchains, compiler generator). Επίσης εξετάζονται θέματα αποσφαλμάτωσης και τεχνικές προγραμματισμού σε επίπεδο kernel, π.χ. signals, probes (non-intrusive monitoring), sysfs/debugs, procfs, interrupt routines, waiting queues/sleeping, syncing, scheduling, timers, kernel memory allocation, kthreads, και tasklets. Επίσης λεπτομερώς εξετάζεται ο σχεδιασμός και η υλοποίηση Linux character, network devices (netdev, netfilter hooks) και TTY drivers. Δίνεται έμφαση σε θέματα σχεδιασμού και υλοποίησης οδηγών πραγματικών συσκευών (μέσω ioremap, register region) και η επικοινωνία με interfaces υψηλού επιπέδου μέσω IOCTL ή GPIO. Εξηγείται θεωρητικά και η οργάνωση block drivers (DMA-based), ιδιαίτερα RAID, PCIe και USB drivers.

Επίσης εξετάζονται θέματα υποστήριξης τεχνικών multitasking καθώς και προγραμμάτων οδηγών σε συστήματα πραγματικού χρόνου (WCET, WCCT). Εξετάζονται θέματα σχεδιασμού και χρονοπρογραμματισμού διεργασιών σε γνωστά λειτουργικά συστήματα πραγματικού χρόνου, π.χ. FreeRTOS και priority-based preemptive ThreadX (σε Renesas πλατφόρμες). Επίσης εξετάζονται πρακτικά ζητήματα προβλεψιμότητας, ασφάλειας (security) και αξιοπιστίας (safety, reliability) μηνυμάτων σε δίκτυα πραγματικού χρόνου που χρησιμοποιούνται σε συστήματα επικοινωνιών, μεταφορών, διαστημικής μηχανικής, και συστημάτων υγείας με αντίστοιχες επιδείξεις, π.χ. CAN bus σε έξυπνα οχήματα.

Τέλος εξετάζεται η χρήση λειτουργικών συστημάτων πραγματικού χρόνου (RTOS) και βιβλιοθηκών νημάτων (thin clients, protothreads) και small stacks (uIP, protosockets, lwIP) σε περιβάλλοντα με περιορισμένες απαιτήσεις (π.χ. boards βασισμένα σε 8- ή 16-bit AVR μικροελεγκτές) Εδώ εξετάζεται λεπτομερώς και ο προγραμματισμός μικρο-συσκευών (adaptors) μέσω περιφερειακών καναλιών (UART-Serial, I2C, SPI, SWI) σε user-level, και η αντίστοιχη υποστήριξη αυτών σε kernel-level.

Εργαστηριακές Ασκήσεις

Στο εργαστηριακό μέρος του μαθήματος οι φοιτητές έχουν τη δυνατότητα πρακτικής εφαρμογής εννοιών της θεωρίας.

Οι φοιτητές έχουν δυνατότητα πρακτικής εφαρμογής των εννοιών της θεωρίας χρησιμοποιώντας ενσωματωμένες πλατφόρμες (Arduino, STM32, AVNET Zedboard, Hardkernel Odroid XU3/4, 96boards Hikey & Dragonboard 410c, Renesas Synergy S7G2) και υβριδικά συστήματα με mezzanine boards (π.χ. DSP audio, sensor boards) διασυνδεδεμένα με διάφορες τεχνολογίες επικοινωνιών, π.χ. Ethernet, wireless, peripheral

buses, και real-time buses (LIN, CAN bus, ή MOST).

(4) ΔΙΔΑΚΤΙΚΕΣ και ΜΑΘΗΣΙΑΚΕΣ ΜΕΘΟΔΟΙ - ΑΞΙΟΛΟΓΗΣΗ

ΤΡΟΠΟΣ ΠΑΡΑΔΟΣΗΣ.	Πρόσωπο με πρόσωπο στην τάξη		
ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ	- Χρήση Τ.Π.Ε. στη διδασκαλία - Χρήση Τ.Π.Ε. στην εργαστηριακή εκπαίδευση - Χρήση Τ.Π.Ε. στην επικοινωνία με τους φοιτητές μέσω της ηλεκτρονικής πλατφόρμας e-class		
ΟΡΓΑΝΩΣΗ ΔΙΔΑΣΚΑΛΙΑΣ	Δραστηριότητα	Φόρτος Εργασίας Εξαμήνου	
	Διαλέξεις	39	
	Demos/Ασκήσεις	26	
	Προσωπική Μελέτη & Προγρ/σμός (σε Server ή Laptop/PC μέσω VM και Embedded Devices)	55	
	Σύνολο Μαθήματος	120	
ΑΞΙΟΛΟΓΗΣΗ ΦΟΙΤΗΤΩΝ	Μέθοδοι αξιολόγησης:		
	1. Γραπτή τελική εξέταση σε θεωρητικά και κυρίως εργαστηριακά προβλήματα (50%) 2. Προγραμματιστικές εργασίες και προφορική εξέταση (50%) Σε περιπτώσεις φοιτητών με αρκετή προγραμματιστική εμπειρία οι παραπάνω εξετάσεις οι προγραμματιστικές εργασίες αντικαθίστανται από εργασία η οποία καλύπτει σε βάθος σημαντικές ενότητες του μαθήματος (systems programming, Linux drivers & kernel modules, RTOS & real-time systems ή limited systems). Σε αυτή την περίπτωση, στο τελευταίο εργαστηριακό μάθημα, γίνεται ανοικτή παρουσίαση και επίδειξη του συστήματος, εργαλείου, πλατφόρμας, ή εφαρμογής που θα υλοποιήσει ο φοιτητής. Τα κριτήρια αξιολόγησης ανακοινώνονται στους φοιτητές κατά την έναρξη του εξαμήνου και βρίσκονται αναρτημένα στην ιστοσελίδα του μαθήματος στο eClass.		

(5) ΣΥΝΙΣΤΩΜΕΝΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Προτεινόμενη Βιβλιογραφία:

- W.R. Stevens and S.A. Rago, "Advanced Programming in the UNIX Environment", 3rd edition, Addison-Wesley (2013), ISBN 978-0321637734.
- J. Corbet, A. Rubini, and G. Kroah-Hartman, "Linux Device Drivers", O' Reilly, 2006, available from <https://lwn.net/Kernel/LDD3>

Λοιπές πηγές:

- Άρθρα από συναφή επιστημονικά περιοδικά και συνέδρια (στα Αγγλικά)
- Υποστήριξη Εργαστηρίου μέσω eclass με σημειώσεις, παραδείγματα με κώδικα και επεξηγήσεις
- Στο εργαστήριο χρησιμοποιούνται διάφορα αναπτυξιακά boards, καθώς και virtual machines για την ασφαλή ανάπτυξη κώδικα σε επίπεδο kernel, που συνοδεύονται από λογισμικό ανοικτού κώδικα ή licensed software και εγχειρίδια