Dentro del marco de la asignatura "Computing Selected Topics"
Dra. Olga Kolesnikova imparte el curso de
"Procesamiento de lenguaje natural (PLN)"

Programa del curso

UNIDAD I. Fundamentos de Procesamiento de Lenguaje Natural (PLN)
1.1. Conceptos básicos lingüísticos usados en PLN
1.2. Retos básicos en el procesamiento de lenguaje por computadora
1.3. Temas y problemas fundamentales en el área de PLN

UNIDAD II. Python como un lenguaje de programación que es conveniente usar para PLN
2.1. Tipos de datos
2.2. Cadenas y sus métodos
2.3. Funciones y módulos
2.4. Manejo de archivos
2.5. Codificación
2.6. Peculiaridades de procesamiento de cadenas en el idioma español a diferencia del idioma inglés (letras con acentos, símbolos de puntuación ausentes en inglés, etc.)

UNIDAD III. Fundamentos matemáticos para PLN.
3.1. Probabilidad
3.2. Verosimilitud
3.3. Entropía
3.4. Información mutual
3.5. Estimación de máxima verosimilitud

UNIDAD IV. Introducción a Natural Language Toolkit (NLTK) como herramienta de PLN en Python
4.1. Instalación
4.2. Carga de corpus (colección de textos)
4.3. Creación de una concordancia
4.4. Detección de los mismos contextos para dos palabras
4.5. Detección de palabras similares
4.6. Creación de distribución de palabras en un texto
4.7. Tokenización y extracción de colocaciones

UNIDAD V. Normalización de texto
5.1. Obtención de texto plano a partir de un archivo HTML
5.2. Eliminación de símbolos de puntuación, dígitos, símbolos especiales con expresiones regulares
5.3. Obtención de tokens
5.4. Obtención de lemas
5.5. Eliminación de palabras auxiliares (stopwords)

UNIDAD VI. Representación vectorial de palabras y medidas de similitud de palabras
6.1. Modelo de bolsa de palabras
6.2. Espacio vectorial para representación de contexto de palabras
6.3. Medición de similitud entre vectores: producto punto
6.4. La métrica tf-idf
6.5. Coseno del ángulo entre vectores
6.6. Extracción de palabras similares utilizando diferentes medidas de similitud

UNIDAD VII. Minería de relaciones sintagmáticas entre palabras
7.1. Segmentación de texto en oraciones
7.2. Entropía para detección de relaciones sintagmáticas
7.3. Información mutual para extracción de palabras relacionadas

UNIDAD VIII. Minería de tópicos en textos de noticias
8.1. Modelo generativo de tópicos
8.2. Optimización de parámetros del modelo generativo de tópicos utilizando el algoritmo de esperanza – maximización
8.3. Representación de tópicos con una palabra clave
8.4. Minería de palabras clave utilizando los métodos de anotación de palabras con partes de oración.
8.5. Minería de sustantivos en textos de noticias y obtención de sustantivos más frecuentes como candidatos de palabras clave.
8.6. Implementación del modelo generativo de tópicos en el paquete “gensim” con el método LDA

UNIDAD IX. Clasificación de textos
9.1. Definición de la tarea de clasificación, su motivación y uso práctico.
9.2. Clasificadores generativos y discriminativos
9.3. Clasificador Bayesiano ingenuo
9.4. Implementación del clasificador Bayesiano ingenuo
9.5. Regresión logística como clasificador discriminativo, su modelo matemático
9.6. Problema de demasiado ajuste (overfitting), los métodos de su resolución: regresión de cresta (regularización L2) y regresión de laso (regularización L1).
9.7. Algoritmo de K vecinos más cercanos, su modelo matemático
9.8. Algoritmo de árbol KD en el clasificador de K vecinos más cercanos
9.9. Implementación del clasificador de K vecinos más cercanos

UNIDAD X. Minería de opinión y detección de polaridad y sentimientos
10.1. Regresión logística ordinal para la minería de opinión.
10.2. El paquete “mord” de Python para la implementación de regresión logística ordinal
10.3. Detección de polaridad utilizando clasificadores de texto.
10.4. Análisis de errores de los clasificadores y posibles estrategias de mejorar el rendimiento de los clasificadores
10.5. Normalización de frecuencias de palabras para mejorar el rendimiento de los clasificadores.
10.6. Medidas de polaridad: binaria y en el rango [-1,1]
10.7. Cálculo de polaridad usando los diccionarios de polaridad con diferentes medidas de polaridad.

10.8. Extracción de aspectos para la detección de polaridad más detallada.
10.9. Extracción de aspectos con unigramas y bigramas más frecuentes utilizando los métodos de secuencias de Python.
10.10. Cálculo de polaridad de los aspectos.

UNIDAD XI. Agrupación de textos (clustering)
11.1. Definición de la tarea de agrupación de textos, su motivación y uso práctico.
11.2. Métodos generativos para la agrupación de textos
11.3. Agrupación jerárquica
11.4. Agrupación no jerárquica
11.5. Agrupación aglomerante
11.6. Método de K promedios para la agrupación de textos.

UNIDAD XII. Generación automática de resumen de texto
12.1. Resumen extractivo y abstractivo.
12.2. Método summarize() del paquete “gensim”.
12.3. Método de TextRank.
12.4. Método LSA.

UNIDAD XIII. Extracción de información
13.1. Fragmentación de texto (la técnica de “chunking” y “chinking”) y extracción de entidades y relaciones entre las entidades.
13.2. Generación de árbol de fragmentos (frases nominales y frases verbales).
13.2. Detección de entidades nombradas

UNIDAD XIII. Tendencias recientes en el procesamiento de lenguaje natural

Bibliografía

1. Zhai, ChengXiang, and Sean Massung. *Text data management and analysis: a practical introduction to information retrieval and text mining*. Morgan & Claypool, 2016.
2. Sarkar, Dipanjan. *Text analytics with Python: A practical real-world approach to gaining actionable insights from your data*. Apress, 2016.
3. Bird, Steven, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.
4. Mitchell, Ryan. *Web Scraping with Python: Collecting More Data from the Modern Web*. " O'Reilly Media, Inc.", 2018.
5. Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.
6. Martin, James H., and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River: Pearson/Prentice Hall, 2009.
7. Manning, Christopher D., Christopher D. Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

SYNTHESIZED SCHOOL PROGRAM

ACADEMIC UNIT: Escuela Superior de Cómputo

ACADEMIC PROGRAM: Ingeniería en Sistemas Computacionales.

LEARNING UNIT: Computing Selected Topics **LEVEL:** III

AIM OF THE LEARNING UNIT:
The student restructures the LINUX operating system kernel, based on their analysis and redesign techniques.

CONTENTS:

I. Introduction.
II. Memory and Process Management.
III. Time management and synchronization.
IV. Devices and file systems.
V. Embedded Linux.

TEACHING PRINCIPLES:
The teacher will apply a Projects-Based learning process, through inductive and heuristic methods so that group members participate in a cooperative and mutual support in learning.
The collaborative learning activities can consist of: brainstorming, discussion groups, cells of learning, structured problem solving, analysis teams and research in groups.
The projects will be proposed by the teacher, and be associated with the core development, which must be designed and programmed in teams.

EVALUATION AND PASSING REQUIREMENTS:

The program will evaluate the students from the portfolio of evidence, which is made up of: formative and summative assessment, using rubrics for self, peer and hetero assessment.

Other means to pass this Unit of Learning:

- Evaluation of acknowledges previously acquired, with base in the issues defined by the academy.
- Official recognition by either another IPN Academic Unit of the IPN or by a national or international external academic institution besides IPN.

REFERENCES:

- Bovet, D. Cesati, M. (2000) Understanding the Linux Kernel. (3ª Ed.) EUA: Ed. O'Reilly. ISBN: 0-596-00002-2
- Hallinan, C. (2006) Embedded Linux Primer. EUA: Ed. Prentice Hall. ISBN: 978-0-13-167984-9
- Kroah-Hartman, G. (2006) Linux Kernel in a Nutshell. EUA: Ed. O'Reilly. ISBN: 978-0-596-10079-7
- Love, R. (2003) Linux Kernel Development. (2a Ed.) EUA: Ed. Novell Press. ISBN: 978-0672327209
- Nutt, G. (2000) Kernel Projects for Linux. EUA: Ed. Addison Wesley. ISBN: 978-0201612431

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

ACADEMIC UNIT: Escuela Superior de Cómputo.
ACADEMIC PROGRAM: Ingeniería en Sistemas Computacionales
LATERAL OUTPUT: Analista Programador de Sistemas de Información.
FORMATION AREA: Professional.
MODALITY: Presence.

LEARNING UNIT: Computing Selected Topics.
TYPE OF LEARNING UNIT: Theorical - Practical, Optative.
VALIDITY: August, 2011.
LEVEL: III.
CREDITS: 7.5 Tepic, 4.39 SATCA

ACADEMIC AIM

This learning unit contributes to the profile of the Engineer in Computer Systems by developing skills as well as creative thinking and programming system software that make optimal use of resources.
It also encourages self learning ability, along with participatory and collaborative attitudes in problem solving and project planning working groups.

This unit has the units Algorithm and Structured Programming, Data Structures and Operating Systems as antecedents. The consequent units are Terminal Work I and II.

AIM OF THE LEARNING UNIT:

The student restructures the LINUX operating system kernel, based on their analysis and redesign techniques.

CREDITS HOURS

THEORETICAL CREDITS / WEEK: 3.0

PRACTICAL CREDITS / WEEK: 1.5

THEORETICAL HOURS / SEMESTER: 54

PRACTICAL HOURS / SEMESTER: 27

AUTONOMOUS LEARNING HOURS: 54

CREDITS HOURS / SEMESTER: 81

LEARNING UNIT DESIGNED BY:
Academia de Sistemas Distribuidos.

REVISED BY:
Dr. Flavio Arturo Sánchez Garfias.
Subdirección Académica

APPROVED BY:
Ing. Apolinar Francisco Cruz Lázaro.
Presidente del CTCE

AUTHORIZED BY: Comisión de Programas Académicos del Consejo General Consultivo del IPN

Ing. Rodrigo de Jesús Serrano Domínguez
Secretario Técnico de la Comisión de Programas Académicos

THEMATIC UNIT: I **TITLE:** Introduction

UNIT OF COMPETENCE

The student carries out the compilation of the Linux operating system kernel, based on the configuration and compilation procedures.

No.	CONTENTS	Teacher led-instruction HOURS		Autonomous Learning HOURS		REFERENCES KEY
		T	P	T	P	
1.1	History of LINUX.	0.5		2.0		
1.2	General architecture of the core.	0.5	0.5		2.5	2B,7B,6C,5C
1.2.1	Core structure.					
1.2.2	Supervisor mode and user mode.					
1.3	Construction of the core.	1.5	0.5	2.0	1.0	
1.3.1	Tools for the kernel.					
1.3.2	Download the source code.					
1.3.3	Tool chain.					
1.3.4	Tools to use.					
1.4	Configuration and compilation.	1.0	0.5	2.0	1.0	
1.4.1	Installing and running the core.					
1.4.2	Kernel upgrade.					
1.4.3	Customizing the kernel.					
1.4.4	Installing device drivers.					
1.5	System Calls.	1.0	0.5	1.5	2.5	
1.5.1	Transfer control.					
1.5.2	Service routines.					
1.5.3	Passing parameters.					
1.5.4	Signal and interrupts.					
1.6	Static and dynamic binding.	0.5	0.5	2.5	1.0	
1.6.1	Symbol tables.					
1.6.2	Modules.					
	Subtotals:	5.0	2.5	9.5	8.0	

TEACHING PRINCIPLES

This Thematic Unit must begin with a framing of the course and the formation of teams. In this unit mainly used the strategy of collaborative learning, and heuristic teaching method. The main techniques used are brainstorming, discussion groups, cells of learning, concept maps, structured problem solving, practical work, and analysis and research in teams.

LEARNING EVALUATION

Diagnostic Test

Project Portfolio:

Worksheet	5%
Research team	20%
Report of Practices	15%
Self-Evaluation Rubrics	5%
Cooperative Evaluation Rubrics	5%
Team Project	35%
Written Learning Evidence	15%

THEMATIC UNIT: II **TITLE:** Memory and Process Management

UNIT OF COMPETENCE

The student modifies the scheduling algorithm of Linux processes, based on the performance of their memory and process manager.

No.	CONTENTS	Teacher led-instruction HOURS		Autonomous Learning HOURS		REFERENCES KEY
		T	P	T	P	
2.1	Memory Management.	0.5	0.5	3.0	2.5	2B, 7B, 6C
2.2	Paging and virtual memory.	0.5				
2.2.1	3-page table levels.					
2.2.2	Page frame.					
2.2.3	Paging algorithms.					
2.3	General-purpose cache.	0.5		1.5		
2.4	Swapping and page cache.	0.5				
2.5	Virtual memory management.	0.5				
2.6	Process Management.	0.5	0.5	2.5	2.5	
2.6.1	Structure to manage the processes.					
2.6.2	Process list.					
2.6.3	The file system / proc.					
2.7	Kernel threads.	0.5		1.5		
2.8	Creating processes and associated resources.	0.5				
2.9	Context switch.	0.5				
2.10	Scheduling processes.	0.5				
	Subtotals:	5.0	1.0	8.5	5.0	

TEACHING PRINCIPLES

In this unit mainly to be used the strategy of collaborative learning, and heuristic teaching method. The main techniques used are: brainstorming, discussion groups, cells of learning, concept maps, structured problem solving, practical work, and analysis and research teams in groups. It must develop a team project for the thematic unit, or a single project for the learning unit to incorporate elements of each topic.

LEARNING EVALUATION

Project Portfolio:

Worksheet	5%
Research team	20%
Report of Practices	15%
Self-Evaluation Rubrics	5%
Cooperative Evaluation Rubrics	5%
Team Project	35%
Written Learning Evidence	15%

THEMATIC UNIT: III **TITLE:** Time management and synchronization

UNIT OF COMPETENCE

The student implements a timer module based on interrupt handling.

No.	CONTENTS	Teacher led-instruction HOURS		Autonomous Learning HOURS		REFERENCES KEY
		T	P	T	P	
3.1	Interrupt handling.	0.5	0.5	1.5	2.5	2B, 7B, 6C
3.2	Upper half and lower half.	0.5				
3.3	Using the tasklet.	0.5				
3.4	Clocks in Linux.	0.5		1.5		
3.4.1	RTC, the CPU cycle counter, the core clock.					
3.4.2	Timer interrupts.					
3.5	Soft IRQ.	0.5				
3.6	Disabling interrupts.	0.5		2.0		
3.7	Spin lock.	1.0				
3.8	Kernel semaphores.					
	Subtotals:	4.0	0.5	5.0	2.5	

TEACHING PRINCIPLES

In this unit mainly to be used the strategy of collaborative learning, and heuristic teaching method. The main techniques used are: brainstorming, discussion groups, cells of learning, concept maps, structured problem solving, practical work, and analysis and research teams in groups. It must develop a team project for the thematic unit, or a single project for the learning unit to incorporate elements of each topic.

LEARNING EVALUATION

Project Portfolio:

Worksheet	5%
Research team	20%
Report of Practices	15%
Self-Evaluation Rubrics	5%
Cooperative Evaluation Rubrics	5%
Team Project	35%
Written Learning Evidence	15%

THEMATIC UNIT: IV **TITLE:** Devices and file systems

UNIT OF COMPETENCE

The student implements a device driver in Linux, based in synchronization mechanisms and performance of interruptions.

No.	CONTENTS	Teacher led-instruction HOURS		Autonomous Learning HOURS		REFERENCES KEY
		T	P	T	P	
4.1	Device driver.	0.5	0.5	1.5	2.5	2B,7B,3C
4.2	DMA.	0.5				
4.3	Character devices, block and network.	0.5				
4.4	Special device file.	0.5		1.5		
4.5	Device registration.	0.5				
4.6	LINUX file systems.	0.5	0.5		1.0	
4.7	File systems.	0.5		1.5		
4.7.1	VFS virtual file systems.					
4.7.2	LVM and RAID.					
4.7.3	JFS.					
	Subtotals:	3.5	1.0	4.5	3.5	

TEACHING PRINCIPLES

In this unit mainly to be used the strategy of collaborative learning, and heuristic teaching method. The main techniques used are: brainstorming, discussion groups, cells of learning, concept maps, structured problem solving, practical work, and analysis and research teams in groups. It must develop a team project for the thematic unit, or a single project for the learning unit to incorporate elements of each topic.

LEARNING EVALUATION

Project Portfolio:

Worksheet	5%
Research team	20%
Report of Practices	15%
Self-Evaluation Rubrics	5%
Cooperative Evaluation Rubrics	5%
Team Project	35%
Written Learning Evidence	15%

THEMATIC UNIT: V **TITLE:** Embedded Linux

UNIT OF COMPETENCE

The student implements an embedded system based on design strategies and the source code for Linux.

No.	CONTENTS	Teacher led-instruction HOURS		Autonomous Learning HOURS		REFERENCES KEY
		T	P	T	P	
5.1	Architecture of an embedded system.	1.5	0.5	2.5	2.5	4C
5.2	Development process of embedded systems.					
5.3	Boot loaders.	1.0				
5.4	Development tools.	1.0		2.5		
5.5	Real time.	0.5				
	Subtotals:	4.0	0.5	5.0	2.5	

TEACHING PRINCIPLES

In this unit mainly to be used the strategy of collaborative learning, and heuristic teaching method. The main techniques used are: brainstorming, discussion groups, cells of learning, concept maps, structured problem solving, practical work, and analysis and research teams in groups. It must develop a team project for the thematic unit, or a single project for the learning unit to incorporate elements of each topic.

LEARNING EVALUATION

Project Portfolio:

Worksheet	5%
Research team	20%
Report of Practices	15%
Self-Evaluation Rubrics	5%
Cooperative Evaluation Rubrics	5%
Team Project	35%
Written Learning Evidence	15%

RECORD OF PRACTICALS

No.	NAME OF THE PRACTICAL	THEMATIC UNITS	DURATION	ACCOMPLISHMENT LOCATION
1	Installing and compiling the Linux kernel.	I	3.0	Computer Labs.
2	Optimization of kernel and driver installation.	I	1.5	
3	Getting kernel variables by accessing the / proc.	I	1.5	
4	Construction of a module.	I	3.0	
5	Incorporating a new system call.	I	1.5	
6	Virtual memory performance.	II	3.0	
7	Modifying the scheduler.	II	3.0	
8	Implementing timers.	III	3.0	
9	Building a device driver.	IV	3.0	
10	File system Optimization.	IV	1.5	
11	Embedded Linux.	V	3.0	
		TOTAL OF HOURS	27.0	

EVALUATION AND PASSING REQUIREMENTS:

The practices are considered mandatory to pass this learning unit.
The practices worth 15% in each thematic unit.

PERIOD	UNIT	EVALUATION TERMS
1	I	Continuous evaluation 85% and written learning evidence 15%
2	II	Continuous evaluation 85% and written learning evidence 15%
	III	Continuous evaluation 85% and written learning evidence 15%
3	IV	Continuous evaluation 85% and written learning evidence 15%
	V	Continuous evaluation 85% and written learning evidence 15%
		The learning unit I is 30% worth of the final score. The learning unit II is 24% worth of the final score. The learning unit III is 15% worth of the final score. The learning unit IV is 16% worth of the final score. The learning unit V is 15% worth of the final score. Other means to pass this Learning Unit: • Evaluation of acknowledges previously acquired, with base in the issues defined by the academy. • Official recognition by either another IPN Academic Unit of the IPN or by a national or international external academic institution besides IPN. If accredited by Special Assessment or a certificate of proficiency, this will be based on guidelines established by the academy on a previous meeting for this purpose.

KEY	B	C	REFERENCES
1		X	Benvenuti, C. (2005).*Understanding Linux Network Internals*. EUA: Ed. O'Reilly Media.ISBN: 978-0596002558.
2	X		Bovet, D. Cesati,M. (2000) *Understanding the Linux Kernel*. (3ª Ed.). EUA: Ed. O'Reilly. ISBN: 0-596-00002-2.
3		X	Corbet, J. (2005).*Linux Device Drivers*. EUA: Ed. O'Reilly Media. ISBN: 978-0596005900.
4		X	Hallinan, C. (2006).*Embedded Linux Primer*. EUA: Ed. Prentice Hall. ISBN: 978-0-13-167984-9.
5		X	Kroah-Hartman, G. (2006).*Linux Kernel in a Nutshell*. EUA: Ed. O'Reilly. ISBN: 978-0-596-10079-7.
6		X	Love, R. (2003).*Linux Kernel Development*. (2a Ed.) EUA: Ed. Novell Press. ISBN: 978-0672327209.
7	X		Nutt, G. (2000).*Kernel Projects for Linux*. EUA: Ed. Addison Wesley. ISBN: 978-0201612431.

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA ACADÉMICA

DIRECCIÓN DE EDUCACIÓN SUPERIOR

TEACHER EDUCATIONAL PROFILE PER LEARNING UNIT

1. GENERAL INFORMATION

ACADEMIC UNIT: Escuela Superior de Cómputo.

ACADEMIC PROGRAM: Ingeniería en Sistemas Computacionales. **LEVEL** III

FORMATION AREA:

Institutional	Basic Scientific	Professional	Terminal and Integration

ACADEMY: Sistemas Distribuidos. **LEARNING UNIT:** Computing Selected Topics.

SPECIALTY AND ACADEMIC REQUIRED LEVEL: Masters Degree or Doctor in Computer Science.

2. AIM OF THE LEARNING UNIT:
The student restructures the LINUX operating system kernel, based on their analysis and redesign techniques.

3. PROFFESSOR EDUCATIONAL PROFILE:

KNOWLEDGE	PROFESSIONAL EXPERIENCE	ABILITIES	APTITUDES
• Computer networks and networked applications. • Handling of the Operating Systems. • Programming Languages C and assembler. • UNIX Systems Programming. • Knowledge of the Institutional Educational Model. • English.	• Experience a year in Linux application programming. • Two years experience in handling groups and collaborative work. • Experience of one year as Professor of Higher Education.	• Analysis and synthesis. • Leadership. • Decision making. • Conflict Management. • Group management. • Verbal fluency of ideas. • Teaching skills.	• Responsible. • Tolerant. • Honest. • Respectful. • Collaborative. • Participative. • Interested to learning. • Assertive.

DESIGNED BY	REVISED BY	AUTHORIZED BY
M. en C. Ukranio Coronilla Contreras COORDINATING PROFESOR M. en C. Jorge Cortés Galicia M. en C. Juan Jesús Gutiérrez García COLLABORATING PROFESSORS	Dr. Flavio Arturo Sánchez Garfias Subdirector Académico	Ing. Apolinar Francisco Cruz Lázaro Director

Date: 2011