

Чернівецький національний університет

імені Юрія Федьковича

“ЗАТВЕРДЖУЮ”

Декан факультету
прикладної математики

(Черевко І.М.)

_____ р. (підпис _____ прізвище, ініціали)

“ _____ ” _____ 20__ р.

РОБОЧА ПРОГРАМА

З курсу _____ *Операційні системи* _____

Для напрямку _____ *6.040301 – Прикладна математика* _____

Факультет _____ *Прикладної математики* _____

Кафедра _____ *Прикладної математики* _____

Форма навчання	Нормативні дані												
	Курс	Семестри	Всього годин	К-сть кредитів	Лекції (год)	Практичних (семінарських) (год)	Лабораторних (год)	Індивідуальна робота (год)	Самостійна робота (год)	Розрахункові, графічні роботи (семестр)	Курсові роботи (семестр)	Заліки (сем)	Екзамен (семестр)
Денна	4	7	90	2,5	17		34		39				7
Заочна													

Зміст програми

Пояснювальна записка до курсу	2
Питання і завдання для самостійної роботи.....	5
Питання для поточного контролю набутих знань	7
Модуль контроль (підсумковий)	12
Тематика лабораторних робіт, їх оцінювання і терміни здачі	13
Критерії оцінювання навчальних досягнень студентів за шкалою ECTS та національною шкалою	19
Рекомендована література	20

Пояснювальна записка до курсу

Семестровий курс „Операційні системи” розроблений з **метою** ознайомлення студентів із фундаментальними принципами проектування і реалізації операційних систем, дозволяє ознайомитися з окремими аспектами функціонування операційних систем на практиці. **Завданням** вивчення дисципліни є сформулювати у студентів уявлення про будову та принципи роботи, стан і перспективи розвитку сучасних операційних систем.

Рекомендований рівень попередньої підготовки – володіння комп'ютером і його внутрішньою будовою на рівні користувача, знання алгоритмічної мови Сі.

Після вивчення курсу студент повинен оволодіти таким теоретичним матеріалом: поняття і реалізація процесів, взаємодія процесів, проблеми взаємоблокування, організація пам'яті, структура файлової системи, система вводу-виводу, мережі і безпека операційних систем. Також повинен здобути практичні навички з таких тем: організація процесів, різні способи їхньої взаємодії, пристрій файлової системи, системи вводу-виводу, основи мережного програмування.

Тематика змістових модулів та їх основні положення

ЗМІСТОВИЙ МОДУЛЬ 1

« Поняття операційної системи. Процеси. Базові аспекти організації, планування та взаємодії процесів. »

НЕ 1.1. Вступ. Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.

Структура обчислювальної системи. Операційна система як віртуальна машина, як менеджер ресурсів, як захисник користувачів і програм, як постійно функціонуюче ядро. Коротка історія еволюції обчислювальних систем. Основні поняття, концепції ОС (Системні виклики, переривання, виняткові ситуації, файли, процеси, нитки). Архітектурні особливості ОС (монолітне ядро, багаторівневі системи (layered systems), віртуальні машини, мікроядерна архітектура, змішані системи). Класифікація ОС (реалізація багатозадачності, підтримка багатокористувачького режиму, багатопроцесорна обробка, системи реального часу).

НЕ 1.2. Процеси. Їх стани і операції над ними.

Поняття процесу. Стани процесу. Операції над процесами і пов'язані з ними поняття (набір операцій, Process Control Block і контекст процесу, одноразові операції, багаторазові операції, перемикання контексту).

НЕ 1.3. Планування процесів.

Рівні планування. Критерії планування і вимоги до алгоритмів планування. Параметри планування. Витісняльне і невитісняльне планування. Алгоритми планування (First-Come, First-Served (FCFS), Round Robin (RR), Shortest-Job-First (SJF), гарантоване планування, пріоритетне планування, багаторівневі черги (Multilevel Queue), багаторівневі черги зі зворотнім зв'язком(Multilevel Feedback Queue)).

НЕ 1.4. Кооперація процесів і основні аспекти її логічної організації.

Процеси, що взаємодіють. Категорії засобів обміну інформацією. Логічна організація механізму передачі інформації (встановлення зв'язку, інформаційна валентність процесів і засобів зв'язку, особливості передачі інформації за допомогою ліній зв'язку (буферизація, потік вводу/виводу і повідомлень), надійність ліній зв'язку, завершення зв'язку). Нитки виконання.

НЕ 1.5. Критичні секції процесів, взаємовиключення і організація правильної черговості. Алгоритми синхронізації процесів

Interleaving, race condition і взаємовиключення. Критична секція. Програмні алгоритми організації взаємодії процесів(вимоги до алгоритмів, заборона переривань, строге чередування, прапорці готовності, алгоритм Петерсона, алгоритм булочної (bakery algorithm)). Апаратна підтримка взаємовиключень (команда Test-and-Set (перевірити і присвоїти 1), команда Swap (обміняти значення)).

ЗМІСТОВИЙ МОДУЛЬ 2

« Розширені аспекти організації, планування та взаємодії процесів. Віртуальна пам'ять та засоби для її організації. »

НЕ 2.1. Семафори, монітори, повідомлення і їх еквівалентність.

Семафори. Концепція семафорів. Вирішення проблеми producer-consumer за допомогою семафорів. Монітори. Повідомлення. Еквівалентність семафорів, моніторів та повідомлень реалізація моніторів і передачі повідомлень за допомогою семафорів, реалізація семафорів і передачі повідомлень за допомогою моніторів, реалізація семафорів і моніторів за допомогою черги повідомлень).

НЕ 2.2. Тупики і боротьба з ними.

Поняття тупика. Умови виникнення тупиків. Основні напрямки боротьби з тупиками. Ігнорування проблеми тупиків. Способи запобігання тупиків шляхом чіткого розподілу ресурсів. Алгоритм банкіра. Запобігання тупиків за рахунок порушення умов виникнення тупиків (порушення умови взаємовиключення, порушення умови очікування додаткових ресурсів, порушення принципу відсутності перерозподілу, порушення умови кругового очікування). Виявлення тупиків. Відновлення після тупиків.

НЕ 2.3. Найпростіші схеми керування пам'яттю.

Фізична організація пам'яті комп'ютера. Основна і вторинна пам'ять. Принцип локальності. Логічна пам'ять. Зв'язування адрес. Функції системи керування пам'яттю. Найпростіші схеми керування пам'яттю(схема з фіксованими розділами, один процес в пам'яті, оверлейна структура, динамічний розподіл, своппінг, схема зі змінними розділами). Сторінкова пам'ять. Сегментна і сегментно-сторінкова організація пам'яті.

НЕ 2.4. Віртуальна пам'ять. Архітектурні засоби підтримки віртуальної пам'яті.

Поняття віртуальної пам'яті. Архітектурні засоби підтримки віртуальної пам'яті. Сторінкова віртуальна пам'ять. Сегментно-сторінкова організація віртуальної пам'яті. Структура таблиці сторінок. Асоціативна пам'ять. Інвертована таблиця сторінок. Розмір сторінки.

НЕ 2.5. Апаратно незалежний рівень керування віртуальною пам'яттю.

Особливі ситуації при роботі з пам'яттю. Стратегії керування сторінковою пам'яттю. Алгоритми заміщення сторінок. Алгоритм FIFO. Виштовхування сторінки, яка прийшла першою. Аномалія Біледі (Belady). Оптималь-

ний алгоритм (OPT). Виштовхування сторінки, яка найдовше не використовувалася. Алгоритм LRU. Виштовхування сторінки, яка рідко використовується. Алгоритм NFU. Інші алгоритми. Керування кількістю сторінок, які виділені процесу. Модель робочої множини. Трешинг (Thrashing). Модель робочої множини. Сторінкові демони. Програмна підтримка сегментної моделі пам'яті процесу. Окремі аспекти функціонування менеджера пам'яті.

ЗМІСТОВИЙ МОДУЛЬ 3

« Файлова система, її реалізація. Пристрої вводу-виводу. Робота з мережею в ОС. Безпека та захист в ОС.»

НЕ 3.1. Файлові системи з точки зору користувача. Файли і операції над ними. Каталоги.

Файлова система. Загальні відомості про файли. Імена файлів. Типи файлів. Атрибути файлів. Організація файлів і доступ до них. Послідовний файл. Файл з прямим доступом. Інші форми організації файлів. Операції над файлами. Директорії. Логічна структура файлового архіва. Розділи диска. Організація доступу до архіва файлів. Операції над директоріями. Захист файлів. Контроль доступу до файлів. Списки прав доступу.

НЕ 3.2. Реалізація файлової системи і каталогів. Способи виділення дискового простору. Розділені файли. Цілісність файлової системи.

Загальна структура файлової системи. Керування зовнішньою пам'яттю. Методи виділення дискового простору (виділення неперервної послідовності блоків, зв'язний список, таблиця відображення файлів, індексні вузли). Керування вільним і зайнятим дисковим простором. Облік за допомогою організації бітового вектора. Облік за допомогою організації зв'язного списку. Розмір блоку. Структура файлової системи на диску. Реалізація каталогів. Приклади реалізації каталогів у деяких ОС (MS-DOS, Unix). Пошук у директорії (лінійний пошук, хеш-таблиця, інші методи пошуку). Монтування файлових систем. Зв'язування файлів. Кооперація процесів при роботі з файлами. Приклади вирішення колізій і тупикових ситуацій. Надійність файлової системи. Цілісність файлової системи (порядок виконання операцій, журналювання, перевірка цілісності файлової системи за допомогою утиліт). Керування "поганими" блоками.

Продуктивність файлової системи (кешування, оптимальне розміщення інформації на диску). Системні виклики, що працюють із символьним ім'ям файлу (системні виклики, що зв'язують pathname з дескриптором файлу; зв'язування файлу; видалення файлу). Системні виклики, що працюють із файловим дескриптором. Функції вводу-виводу з файлу. Сучасні архітектури файлових систем.

НЕ 3.3. Пристрої вводу-виводу. Апарат переривань. Завдання системи вводу-виводу. Блочні і символьні пристрої. Алгоритми вибору наступного запиту для диску.

Загальні відомості про архітектуру комп'ютера. Структура контролера пристрою. Опитування пристроїв і переривання. Виняткові ситуації і системні виклики. Прямий доступ до пам'яті (Direct Memory Access - DMA). Логічні принципи організації вводу-виводу. Структура системи вводу-виводу. Систематизація зовнішніх пристроїв і інтерфейс між базовою підсистемою вводу-виводу і драйверами. Функції базової підсистеми вводу-виводу (системні виклики, що блокуються; системні виклики, що не блокуються; асинхронні системні виклики; буферизація і кешування; spooling і захоплення пристроїв; обробка переривань і помилок; планування запитів). Алгоритми планування запитів до жорсткого диска (будова жорсткого диска і параметри планування; алгоритм First Come First Served (FCFS); алгоритм Short Seek Time First (SSTF); алгоритми сканування (SCAN, C-SCAN, LOOK, C-LOOK)).

НЕ 3.4. Мережеві операційні системи.

Мережні і розподілені операційні системи. Взаємодія віддалених процесів як основа роботи обчислювальних мереж. Основні питання логічної організації передачі інформації між віддаленими процесами. Поняття протоколу. Багаторівнева модель побудови мережних обчислювальних систем. Проблеми адресації в мережі (однорівневі адреси; дворівневі адреси; віддалена адресація і встановлення адрес; локальна адресація, поняття порту; повні адреси; поняття сокета (socket)). Проблеми маршрутизації в мережах. Зв'язок із встановленням логічного з'єднання і передача даних за допомогою повідомлень. Синхронізація віддалених процесів.

НЕ 3.5. Основні проблеми інформаційної безпеки.

Загрози безпеки. Формалізація підходу до забезпечення інформаційної безпеки. Криптографія як одна з базових технологій безпеки ОС. Шифрування з використанням алгоритму RSA. Теорема Ейлера.

НЕ 3.6. Захисні механізми операційних систем.

Ідентифікація і аутентифікація. Паролі, вразливість паролів. Шифрування пароля. Авторизація. Розмежування доступу до об'єктів ОС. Домени безпеки. Матриця доступу (список прав доступу, access control list; мандати можливостей, capability list; інші способи контролю доступу; зміна домена). Неприпустимість повторного використання об'єктів. Виявлення вторгнень. Аудит системи захисту. Аналіз деяких популярних ОС із погляду їхньої захищеності (MS-DOS, NetWare, IntranetWare, OS/2, Unix, Windows NT/2000/XP).

Питання і завдання для самостійної роботи

№	Назва тем та зміст питань	К-сть годин
1.	Вступ до курсу практичних занять. Знайомство з операційною системою UNIX. Вступ до курсу практичних занять. Коротка історія операційної системи UNIX, її структура. Системні виклики та бібліотека libc. Поняття login і password. Спрощене поняття про будову файлової системи в UNIX. Повні імена файлів. Поняття про поточний каталог. Команда pwd. Відносні імена файлів. Домашній каталог користувача. Команда map – універсальний довідник. Команди cd – зміни поточного каталогу і ls – перегляду вмісту каталогу. Команда cat і створення файлу. Пере-напрявлення вводу/виводу. Найпростіші команди для роботи з файлами – cp, mv, mkdir, mv. Історія редагування файлів – ed, vi. Система Midnight Commander – mc. Вбудований в mc редактор і редактор joe. Користувач і група. Команди chown і chgrp. Права доступу до файлу. Команда ls з опціями -al. Використання команд chmod і umask. Системні виклики getuid і getgid. Компіляція програм мовою C в UNIX і запуск їх на виконання. [1, 2, 3, 4, 5, 6]	4
2.	Процеси в операційній системі UNIX. Поняття процесу в UNIX, його контекст. Ідентифікація процесу. Стан процесу. Коротка діаграма станів. Ієрархія процесів. Системні виклики getpid(), getppid(). Створення процесу в UNIX. Системний виклик fork(). Завершення процесу. Функція exit(). Параметри функції main() у мові C. Змінні середовища та аргументи командного рядка. Зміна користувальницького контексту процесу. Сімейство функцій для системного виклику exec(). [1, 3, 7]	4
3.	Організація взаємодії процесів через pipe і FIFO в UNIX. Поняття потоку вводу-виводу. Уявлення про роботу з файлами через системні виклики і стандартну бібліотеку вводу-виводу. Поняття файлового дескриптора. Відкриття файлу. Системний виклик open(). Системні виклики close(), read(), write(). Поняття pipe. Системний виклик pipe(). Організація зв'язку через pipe між процесом-батьком і процесом-нащадком. Успадкування файлових дескрипторів при викликах fork() і exec(). Особливості поводження викликів read() і write() для pip'a. Поняття FIFO. Використання системного виклику mknod() для створення FIFO. Функція mkfifo(). Особливості поводження виклику open() при відкритті FIFO. [1, 3, 7]	5
4.	Засоби System V IPC. Організація роботи з розділюваною пам'яттю в UNIX. Поняття ниток виконання (thread). Переваги та недоліки потокового обміну даними. Поняття System V IPC. Простір імен. Адресація в System V IPC. Функція flock(). Дескриптори System V IPC. Розділювана пам'ять в UNIX. Системні виклики shmget(), shmat(), shmdt(). Команди ipc і ipcrm. Використання системного виклику shmctl() для звільнення ресурсу. Розділювана пам'ять і системні виклики fork(), exec() і функція exit(). Поняття про нитки виконання (thread) в UNIX. Ідентифікатор нитки виконання. Функція pthread_self(). Створення і завершення thread'a. Функції pthread_create(), pthread_exit(), pthread_join(). Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять. [1, 3, 7]	5
5.	Семафори в UNIX як засіб синхронізації процесів. Семафори в UNIX. Відмінність операцій над UNIX-семафорами від класичних операцій. Створення масиву семафорів або доступ до вже існуючого масиву. Системний виклик semget(). Виконання операцій над семафорами. Системний виклик semop(). Видалення набору семафорів із системи за допомогою команди ipcrm або системного виклику semctl(). Поняття про POSIX-семафори. [1, 3, 7]	4
6.	Черги повідомлень в UNIX. Повідомлення як засоби зв'язку і засіб синхронізації процесів. Черги повідомлень в UNIX як складова частина System V IPC. Створення черги повідомлень або доступ до вже існуючої. Системний виклик msgget(). Реалізація примітивів send і receive. Системні виклики msgsnd() і msgrcv(). Видалення черги повідомлень із системи за допомогою команди ipcrm або системного виклику msgctl(). Поняття мультиплексування. Мультиплексування повідомлень. Модель взаємодії процесів клієнт-сервер. Нерівноправність клієнта і сервера. Використання черг повідомлень для синхронізації роботи процесів. [1, 3, 7]	4
7.	Організація файлової системи в UNIX. Робота з файлами та директоріями. Поняття про метому mapped файли. Розділи носія інформації (partitions) в UNIX. Логічна структура файлової системи і типи файлів в UNIX. Організація файлу на диску в UNIX на прикладі файлової си-	4

	<p>стеми s5fs. Поняття індексного вузла (inode). Організація директорій (каталогів) в UNIX. Поняття суперблоку. Операції над файлами та директоріями. Системні виклики та команди для виконання операцій над файлами і директоріями. Системний виклик open(). Системний виклик close(). Операція створення файлу. Системний виклик creat(). Операція читання атрибутів файлу. Системні виклики stat(), fstat() і lstat(). Операції зміни атрибутів файлу. Операції читання з файлу і запису у файл. Операція зміни покажчика поточної позиції. Системний виклик lseek(). Операція додавання інформації у файл. Прапорець O_APPEND. Операції створення зв'язків. Команда ln, системні виклики link() і symlink(). Операція видалення зв'язків і файлів. Системний виклик unlink(). Спеціальні функції для роботи із вмістом директорій. Поняття про файли, відображувані в пам'ять (memoгу mapped файли). Системні виклики mmap(), munmap(). [1, 3, 7]</p>	
8.	<p>Організація вводу-виводу в UNIX. Файли пристроїв. Апарат переривань. Сигнали в UNIX. Поняття віртуальної файлової системи. Операції над файловими системами. Монтування файлових систем. Блокові, символьні пристрої. Поняття драйвера. Блокові, символьні драйвери, драйвери низького рівня. Файловий інтерфейс. Апаратні переривання (interrupt), виключення (exception), програмні переривання (trap, software interrupt). Їхня обробка. Поняття сигналу. Способи виникнення сигналів і види їхньої обробки. Поняття групи процесів, сеансу, лідера групи, лідера сеансу, керуючого терміналу сеансу. Системні виклики getpgrp(), setpgrp(), getpgid(), setpgid(), getsid(), setsid() Системний виклик kill() і команда kill(). Системний виклик signal(). Встановлення власного обробника сигналу. Відновлення попередньої реакції на сигнал. Сигнали SIGUSR1 і SIGUSR2. Використання сигналів для синхронізації процесів. Завершення породженого процесу. Системний виклик waitpid(). Сигнал SIGCHLD. Виникнення сигналу SIGPIPE при спробі запису в pipe або FIFO, який ніхто не збирається читати. Поняття про надійність сигналів. POSIX-функції для роботи із сигналами. [1, 3, 7]</p>	4
9.	<p>Сім'я протоколів TCP/IP. Сокети (sockets) в UNIX і основи роботи з ними. Коротка історія сім'ї протоколів TCP/IP. Загальні відомості про архітектуру сім'ї протоколів TCP/IP. Рівень мережного інтерфейсу. Рівень Internet. Протоколи IP, ICMP, ARP, RARP. Internet-адреси. Транспортний рівень. Протоколи TCP і UDP. UDP і TCP сокети (sockets). Адресні простори портів. Поняття encapsulation. Рівень додатків/програм. Використання моделі клієнт-сервер при вивченні мережного програмування. Організація зв'язку між віддаленими процесами за допомогою датаграм. Мережний порядок байт. Функції htons(), htonl(), ntohs(), ntohl(). Функції перетворення IP-адрес inet_ntoa(), inet_aton(). Функція bzero(). Системні виклики socket(), bind(), sendto(), recvfrom(). Організація зв'язку між процесами за допомогою встановлення логічного з'єднання. Системні виклики connect(), listen(), accept(). Використання інтерфейсу сокетів для інших сімей протоколів. Файли типу "сокет". [1, 4, 8, 9, 10]</p>	5
	Всього годин на самостійну роботу	39

Питання для поточного контролю набутих знань

Усні та письмові опитування не виходять за межі матеріалу, розглянутого на лекційних заняттях, опрацьованого самостійно і на лабораторних заняттях.

Модуль 1

1. Структура обчислювальної системи.
2. Операційна система як віртуальна машина, як менеджер ресурсів, як захисник користувачів і програм, як постійно функціонуюче ядро.
3. Коротка історія еволюції обчислювальних систем. Основні поняття, концепції ОС (Системні виклики, переривання, виняткові ситуації, файли, процеси, нитки).
4. Архітектурні особливості ОС (монолітне ядро, багаторівневі системи (layered systems), віртуальні машини, мікроядерна архітектура, змішані системи).
5. Класифікація ОС (реалізація багатозадачності, підтримка багатокористувацького режиму, багатопроесорна обробка, системи реального часу).
6. Поняття процесу. Стани процесу. Операції над процесами і пов'язані з ними поняття (набір операцій, Process Control Block і контекст процесу, одноразові операції, багаторазові операції, перемикання контексту).
7. Рівні планування. Критерії планування і вимоги до алгоритмів планування.
8. Параметри планування. Витісняльне і невитісняльне планування.
9. Алгоритми планування (First-Come, First-Served (FCFS), Round Robin (RR), Shortest-Job-First (SJF), гарантоване планування, пріоритетне планування, багаторівневі черги (Multilevel Queue), багаторівневі черги зі зворотнім зв'язком).
10. Процеси, що взаємодіють. Категорії засобів обміну інформацією.
11. Логічна організація механізму передачі інформації (встановлення зв'язку, інформаційна валентність процесів і засобів зв'язку, особливості передачі інформації за допомогою ліній зв'язку (буферизація, потік вводу/виводу і повідомлень), надійність ліній зв'язку, завершення зв'язку). Нитки виконання.
12. Interleaving, race condition і взаємовиключення.
13. Критична секція. Програмні алгоритми організації взаємодії процесів (вимоги до алгоритмів, заборона переривань, строге чередування, прапорці готовності, алгоритм Петерсона, алгоритм булочної (bakery algorithm)).
14. Апаратна підтримка взаємовиключень (команда Test-and-Set (перевірити і присвоїти 1), команда Swap (обміняти значення))
15. Коротка історія операційної системи UNIX, її структура.
16. Системні виклики та бібліотека libc. Поняття login і password.
17. Спрощене поняття про будову файлової системи в UNIX. Повні імена файлів. Поняття про поточний каталог. Команда pwd. Відносні імена файлів. Домашній каталог користувача.
18. Команда man – універсальний довідник.
19. Команди cd – зміни поточного каталогу і ls – перегляду вмісту каталогу. Команда cat і створення файлу. Перенапрявлення вводу/виводу. Найпростіші команди для роботи з файлами – cp, rm, mkdir, mv.
20. Історія редагування файлів – ed, vi. Система Midnight Commander – mc. Вбудований в mc редактор і редактор joe.
21. Користувач і група. Команди chown і chgrp. Права доступу до файлу. Команда ls з опціями -al. Використання команд chmod і umask.
22. Системні виклики getuid і getgid. Компіляція програм мовою C в UNIX і запуск їх на виконання.

23. Поняття процесу в UNIX, його контекст. Ідентифікація процесу. Стан процесу. Коротка діаграма станів.
24. Ієрархія процесів. Системні виклики `getpid()`, `getppid()`. Створення процесу в UNIX. Системний виклик `fork()`. Завершення процесу. Функція `exit()`.
25. Параметри функції `main()` у мові C. Змінні середовища та аргументи командного рядка.
26. Зміна користувальницького контексту процесу.
27. Сімейство функцій для системного виклику `exec()`.
28. Поняття потоку вводу-виводу. Робота з файлами через системні виклики і стандартну бібліотеку вводу-виводу. Поняття файлового дескриптора. Відкриття файлу. Системний виклик `open()`. Системні виклики `close()`, `read()`, `write()`.
29. Поняття `pipe`. Системний виклик `pipe()`. Організація зв'язку через `pipe` між процесом-батьком і процесом-нащадком. Успадкування файлових дескрипторів при викликах `fork()` і `exec()`. Особливості поводження викликів `read()` і `write()` для `pipe`'а.
30. Поняття FIFO. Використання системного виклику `mknod()` для створення FIFO. Функція `mkfifo()`. Особливості поводження виклику `open()` при відкритті FIFO. Переваги та недоліки потокового обміну даними.

Модуль 2

1. Семафори. Концепція семафорів. Вирішення проблеми `producer-consumer` за допомогою семафорів. Монітори. Повідомлення. Еквівалентність семафорів, моніторів та повідомлень, реалізація моніторів і передачі повідомлень за допомогою семафорів, реалізація семафорів і передачі повідомлень за допомогою моніторів, реалізація семафорів і моніторів за допомогою черги повідомлень).
2. Поняття тупика. Умови виникнення тупиків. Основні напрямки боротьби з тупиками. Ігнорування проблеми тупиків. Способи запобігання тупиків шляхом чіткого розподілу ресурсів. Алгоритм банкіра. Запобігання тупиків за рахунок порушення умов виникнення тупиків (порушення умови взаємовиключення, порушення умови очікування додаткових ресурсів, порушення принципу відсутності перерозподілу, порушення умови кругового очікування). Виявлення тупиків. Відновлення після тупиків.
3. Фізична організація пам'яті комп'ютера. Основна і вторинна пам'ять. Принцип локальності. Логічна пам'ять. Зв'язування адрес. Функції системи керування пам'яттю. Найпростіші схеми керування пам'яттю (схема з фіксованими розділами, один процес в пам'яті, оверлейна структура, динамічний розподіл, своппінг, схема зі змінними розділами). Сторінкова пам'ять. Сегментна і сегментно-сторінкова організація пам'яті
4. Поняття віртуальної пам'яті. Архітектурні засоби підтримки віртуальної пам'яті. Сторінкова віртуальна пам'ять. Сегментно-сторінкова організація віртуальної пам'яті. Структура таблиці сторінок. Асоціативна пам'ять. Інвертована таблиця сторінок. Розмір сторінки.
5. Особливі ситуації при роботі з пам'яттю. Стратегії керування сторінковою пам'яттю. Алгоритми заміщення сторінок. Алгоритм FIFO. Виштовхування сторінки, яка прийшла першою. Аномалія Біледі (Belady). Оптимальний алгоритм (OPT). Виштовхування сторінки, яка найдовше не використовувалася. Алгоритм LRU. Виштовхування сторінки, яка рідко використовується. Алгоритм NFU. Інші алгоритми. Керування кількістю сторінок, які виділені процесу. Модель робочої множини. Трешинг (Thrashing). Модель робочої множини. Сторінкові демони. Програмна підтримка сегментної моделі пам'ятс процесу. Окремі аспекти функціонування менеджера пам'яті.
6. Поняття System V IPC. Простір імен. Адресація в System V IPC. Функція `ftok()`. Дескриптори System V IPC.
7. Розділювана пам'ять в UNIX. Системні виклики `shmget()`, `shmat()`, `shmdt()`. Команди `ipc` і `ipcrm`. Використання системного виклику `shmctl()` для звільнення ресурсу. Розділювана пам'ять і системні виклики `fork()`, `exec()` і функція `exit()`.

8. Поняття про нитки виконання (thread) в UNIX. Ідентифікатор нитки виконання. Функція pthread_self(). Створення і завершення thread'a. Функції pthread_create(), pthread_exit(), pthread_join(). Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять.
9. Семафори в UNIX. Відмінність операцій над UNIX-семафорами від класичних операцій. Створення масиву семафорів або доступ до вже існуючого масиву. Системний виклик semget(). Виконання операцій над семафорами. Системний виклик semop(). Видалення набору семафорів із системи за допомогою команди ipcrm або системного виклику semctl(). Поняття про POSIX-семафори.
10. Повідомлення як засоби зв'язку і засоб синхронізації процесів. Черги повідомлень в UNIX як складова частина System V IPC. Створення черги повідомлень або доступ до вже існуючої. Системний виклик msgget(). Реалізація примітивів send і receive. Системні виклики msgsnd() і msgrecv(). Видалення черги повідомлень із системи за допомогою команди ipcrm або системного виклику msgctl().
11. Поняття мультиплексування. Мультиплексування повідомлень. Модель взаємодії процесів клієнт-сервер. Нерівноправність клієнта і сервера. Використання черг повідомлень для синхронізації роботи процесів.

Модуль 3

1. Файлова система. Загальні відомості про файли. Імена файлів. Типи файлів. Атрибути файлів. Організація файлів і доступ до них. Послідовний файл. Файл з прямим доступом. Інші форми організації файлів. Операції над файлами. Директорії. Логічна структура файлового архіва. Розділи диска. Організація доступу до архіва файлів. Операції над директоріями. Захист файлів. Контроль доступу до файлів. Списки прав доступу.
2. Загальна структура файлової системи. Керування зовнішньою пам'яттю. Методи виділення дискового простору (виділення неперервної послідовності блоків, зв'язний список, таблиця відображення файлів, індексні вузли). Керування вільним і зайнятим дисковим простором. Облік за допомогою організації бітового вектора. Облік за допомогою організації зв'язного списку. Розмір блоку. Структура файлової системи на диску. Реалізація каталогів. Приклади реалізації каталогів у деяких ОС (MS-DOS, Unix). Пошук у директорії (лінійний пошук, хеш-таблиця, інші методи пошуку). Монтування файлових систем. Зв'язування файлів. Кооперація процесів при роботі з файлами. Приклади вирішення колізій і тупикових ситуацій. Надійність файлової системи. Цілісність файлової системи (порядок виконання операцій, журналювання, перевірка цілісності файлової системи за допомогою утиліт). Керування "поганими" блоками.
3. Продуктивність файлової системи (кешування, оптимальне розміщення інформації на диску). Системні виклики, що працюють із символьним ім'енем файлу (системні виклики, що зв'язують pathname з дескриптором файлу; зв'язування файлу; видалення файлу). Системні виклики, що працюють із файловим дескриптором. Функції вводу-виводу з файлу. Сучасні архітектури файлових систем
4. Загальні відомості про архітектуру комп'ютера. Структура контролера пристрою. Опитування пристроїв і переривання. Виняткові ситуації і системні виклики. Прямий доступ до пам'яті (Direct Memory Access - DMA). Логічні принципи організації вводу-виводу. Структура системи вводу-виводу. Систематизація зовнішніх пристроїв і інтерфейс між базовою підсистемою вводу-виводу і драйверами. Функції базової підсистеми вводу-виводу (системні виклики, що блокуються; системні виклики, що не блокуються; асинхронні системні виклики; буферизація і кешування; spooling і захоплення пристроїв; обробка переривань і помилок; планування запитів). Алгоритми планування запитів до жорсткого диска (будова жорсткого диска і параметри планування; алгоритм First Come First Served (FCFS); алгоритм Short Seek Time First (SSTF); алгоритми сканування (SCAN, C-SCAN, LOOK, C-LOOK)).

5. Мережні і розподілені операційні системи. Взаємодія віддалених процесів як основа роботи обчислювальних мереж. Основні питання логічної організації передачі інформації між віддаленими процесами. Поняття протоколу. Багаторівнева модель побудови мережних обчислювальних систем. Проблеми адресації в мережі (однорівневі адреси; дворівневі адреси; віддалена адресація і встановлення адрес; локальна адресація, поняття порту; повні адреси; поняття сокета (socket)). Проблеми маршрутизації в мережах. Зв'язок із встановленням логічного з'єднання і передача даних за допомогою повідомлень. Синхронізація віддалених процесів.
6. Загрози безпеки. Формалізація підходу до забезпечення інформаційної безпеки. Криптографія як одна з базових технологій безпеки ОС. Шифрування з використанням алгоритму RSA. Теорема Ейлера.
7. Розділи носія інформації (partitions) в UNIX. Логічна структура файлової системи і типи файлів в UNIX. Організація файлу на диску в UNIX на прикладі файлової системи s5fs. Поняття індексного вузла (inode).
8. Організація директорій (каталогів) в UNIX. Поняття суперблоку. Операції над файлами та директоріями. Системні виклики та команди для виконання операцій над файлами і директоріями.
9. Системний виклик open(). Системний виклик close(). Операція створення файлу. Системний виклик creat(). Операція читання атрибутів файлу. Системні виклики stat(), fstat() і lstat(). Операції зміни атрибутів файлу. Операції читання з файлу і запису у файл. Операція зміни покажчика поточної позиції. Системний виклик lseek(). Операція додавання інформації у файл. Прапорці O_APPEND.
10. Операції створення зв'язків. Команда ln, системні виклики link() і symlink(). Операція видалення зв'язків і файлів. Системний виклик unlink(). Спеціальні функції для роботи із вмістом директорій.
11. Поняття про файли, відображувані в пам'ять (memoгу mapped файли). Системні виклики mmap(), munmap().
12. Поняття віртуальної файлової системи. Операції над файловими системами. Монтування файлових систем. Блокові, символні пристрої. Поняття драйвера. Блокові, символні драйвери, драйвери низького рівня. Файловий інтерфейс.
13. Апаратні переривання (interrupt), виключення (exception), програмні переривання (trap, software interrupt). Їхня обробка. Поняття сигналу. Способи виникнення сигналів і види їхньої обробки.
14. Поняття групи процесів, сеансу, лідера групи, лідера сеансу, керуючого терміналу сеансу. Системні виклики getpgrp(), setpgrp(), getpgid(), setpgid(), getsid(), setsid(). Системний виклик kill() і команда kill().
15. Системний виклик signal(). Встановлення власного обробника сигналу. Відновлення попередньої реакції на сигнал. Сигнали SIGUSR1 і SIGUSR2. Використання сигналів для синхронізації процесів.
16. Завершення породженого процесу. Системний виклик waitpid(). Сигнал SIGCHLD. Виникнення сигналу SIGPIPE при спробі запису в pipe або FIFO, який ніхто не збирається читати. Поняття про надійність сигналів. POSIX-функції для роботи із сигналами.
17. Коротка історія сім'ї протоколів TCP/IP. Загальні відомості про архітектуру сім'ї протоколів TCP/IP. Рівень мережного інтерфейсу. Рівень Internet. Протоколи IP, ICMP, ARP, RARP. Internet-адреси. Транспортний рівень. Протоколи TCP і UDP. UDP і TCP сокети (sockets). Адресні простори портів. Поняття encapsulation. Рівень додатків/програм. Використання моделі клієнт-сервер при вивченні мережного програмування.
18. Організація зв'язку між віддаленими процесами за допомогою датаграм. Мережний порядок байт. Функції htons(), htonl(), ntohs(), ntohl(). Функції перетворення IP-адрес inet_ntoa(), inet_aton(). Функція bzero(). Системні виклики socket(), bind(), sendto(), recvfrom().

19. Організація зв'язку між процесами за допомогою встановлення логічного з'єднання. Системні виклики `connect()`, `listen()`, `accept()`. Використання інтерфейсу сокетів для інших сімей протоколів. Файли типу "сокет".

Модуль контроль (підсумковий)

Модуль контроль складається з двох теоретичних питань і двох задач (тематика задач не виходить за межі завдань, що розглядалися та були пророблені на лабораторних заняттях).

1. Еволюція обчислювальних систем.
2. Основні функції операційних систем і принципи їхньої побудови.
3. Процеси. Їхні стани і операції над ними.
4. Планування процесів.
5. Кооперація процесів і основні аспекти її логічної організації.
6. Критичні секції процесів, взаємовиключення і організація правильної черговості.
7. Алгоритми синхронізації процесів.
8. Семафори, монітори, повідомлення і їх еквівалентність.
9. Тупики і боротьба з ними.
10. Найпростіші схеми керування пам'яттю.
11. Віртуальна пам'ять.
12. Архітектурні засоби підтримки віртуальної пам'яті.
13. Апаратно незалежний рівень керування віртуальною пам'яттю.
14. Файлові системи з точки зору користувача.
15. Файли і операції над ними. Каталоги.
16. Реалізація файлової системи і каталогів. Способи виділення дискового простору. Розділювані файли. Цілісність файлової системи.
17. Пристрої вводу-виводу. Апарат переривань.
18. Завдання системи вводу-виводу. Блочні і символічні пристрої. Алгоритми вибору наступного запиту для диску.
19. Мережеві операційні системи.
20. Основні проблеми інформаційної безпеки.
21. Захисні механізми операційних систем.
22. Основні команди операційної системи Unix (Linux).
23. Процеси в операційній системі UNIX.
24. Організація взаємодії процесів через pipe і FIFO в UNIX.
25. Засоби System V IPC. Організація роботи з розділюваною пам'яттю в UNIX. Поняття ниток виконання (thread).
26. Семафори в UNIX як засіб синхронізації процесів.
27. Черги повідомлень в UNIX.
28. Організація файлової системи в UNIX. Робота з файлами та директоріями. Поняття про memory mapped файли.
29. Організація вводу-виводу в UNIX. Файли пристроїв. Апарат переривань. Сигнали в UNIX.
30. Сім'я протоколів TCP/IP. Сокети (sockets) в UNIX і основи роботи з ними.

Тематика лабораторних робіт, їх оцінювання і терміни здачі

№	Тема	Кількість годин	Кількість балів	Термін здачі (№ заняття)
1	Знайомство з ОС Linux.	4	5	2
2	Процеси в ОС Linux.	4	5	4
3	Організація взаємодії процесів через р-іре і FIFO в Linux.	2	5	5
	<i>Контрольна модульна робота</i>		5	
	<i>Модуль 1</i>		20	6
4	Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).	4	7	8
5	Семафори в Linux як засіб синхронізації процесів.	4	6	9
6	Черги повідомлень в Linux.	4	7	11
	<i>Контрольна модульна робота</i>		10	
	<i>Модуль 2</i>		30	12
7	Організація файлової системи в Linux. Робота з файлами та директоріями. Поняття про memory mapped файли.	4	6	13
8	Організація вводу-виводу в Linux. Файли пристроїв. Апарат переривань. Сигнали в Linux.	4	6	15
9	Сім'я протоколів TCP/IP. Сокети (sockets) в Linux і основи роботи з ними.	4	8	17
	<i>Модуль 3</i>		20	17
	<i>Всього за модулі</i>		70	
	<i>Модуль контроль (підсумковий)</i>		30	
	<i>Всього</i>		100	

Лабораторна робота №1

Тема: Знайомство з операційною системою Linux.

1. Вступ до курсу практичних занять.
2. Коротка історія операційної системи UNIX, її структура.
3. Системні виклики та бібліотека libc.
4. Поняття login і password.
5. Спрощене поняття про будову файлової системи в UNIX.
6. Повні імена файлів.
7. Поняття про поточний каталог. Команда pwd.
8. Відносні імена файлів.
9. Домашній каталог користувача.
10. Команда man – універсальний довідник.
11. Команди cd – зміни поточного каталогу і ls – перегляду вмісту каталогу.
12. Команда cat і створення файлу.
13. Перенапрявлення вводу/виводу.
14. Найпростіші команди для роботи з файлами – cp, rm, mkdir, mv.
15. Історія редагування файлів – ed, vi.
16. Система Midnight Commander – mc.
17. Вбудований в mc редактор і редактор joe.
18. Користувач і група. Команди chown і chgrp.
19. Права доступу до файлу.
20. Команда ls з опціями -al.
21. Використання команд chmod і umask.
22. Системні виклики getuid і getgid.
23. Компіляція програм мовою C в UNIX і запуск їх на виконання.

Лабораторна робота №2

Тема: Процеси в операційній системі Linux

1. Поняття процесу в UNIX, його контекст.
2. Ідентифікація процесу.
3. Стан процесу.
4. Коротка діаграма станів. Ієрархія процесів.
5. Системні виклики getpid(), getppid().
6. Створення процесу в UNIX. Системний виклик fork().
7. Завершення процесу. Функція exit().
8. Параметри функції main() у мові C.
9. Змінні середовища та аргументи командного рядка.
10. Зміна користувальницького контексту процесу.
11. Сімейство функцій для системного виклику exec().
12. Виконання програми з fork() з однаковою роботою батька (предка) та дитини (нащадка).
13. Написання, компіляція і запуск програми з використанням виклику fork() з різним поводженням процесів дитини та батька.
14. Написання, компіляція і запуск програми, що роздруковує аргументи командного рядка і параметри середовища.
15. Написання, компіляція і запуск програми для зміни користувачького контексту в породженому процесі.

16. Написання, компіляція і запуск програми з використанням виклику `fork()` з різним поведінням процесів дитини та батька так, щоб породжений процес запускав на виконання нову (довільну) програму.

Лабораторна робота №3

Тема: Організація взаємодії процесів через pipe і FIFO в Linux

1. Поняття потоку вводу-виводу.
2. Уявлення про роботу з файлами через системні виклики і стандартну бібліотеку вводу-виводу.
3. Поняття файлового дескриптора. Відкриття файлу. Системний виклик `open()`. Системні виклики `close()`, `read()`, `write()`.
4. Поняття pipe. Системний виклик `pipe()`. Організація зв'язку через pipe між процесом-батьком і процесом-нащадком.
5. Успадкування файлових дескрипторів при викликах `fork()` і `exec()`.
6. Особливості поведінки викликів `read()` і `write()` для pipe'a.
7. Поняття FIFO. Використання системного виклику `mknod()` для створення FIFO. Функція `mkfifo()`. Особливості поведінки виклику `open()` при відкритті FIFO.
8. Написання, компіляція і запуск програми для читання і запису інформації у файл
9. Програма для pipe в одному процесі.
10. Програма для організації однонапрявленого зв'язку між рідними процесами через pipe.
11. Написання, компіляція і запуск програми для організації двонаправленого зв'язку між родинними процесами через pipe.
12. Програма з FIFO у рідних процессах.
13. Написання, компіляція і запуск програми з FIFO у неспоріднених процессах.

Лабораторна робота №4

Тема: Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread)

1. Переваги та недоліки потокового обміну даними. Поняття System V IPC.
2. Простір імен. Адресація в System V IPC. Функція `ftok()`.
3. Дескриптори System V IPC.
4. Розділювана пам'ять в UNIX. Системні виклики `shmget()`, `shmat()`, `shmdt()`.
5. Команди `ipc` і `ipcrm`.
6. Використання системного виклику `shmctl()` для звільнення ресурсу.
7. Розділювана пам'ять і системні виклики `fork()`, `exec()` і функція `exit()`.
8. Поняття про нитки виконання (thread) в UNIX.
9. Ідентифікатор нитки виконання.
10. Функція `pthread_self()`.
11. Створення і завершення thread'a.
12. Функції `pthread_create()`, `pthread_exit()`, `pthread_join()`.
13. Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять.
14. Програма з використанням розділюваної пам'яті
15. Програма з використанням двох ниток виконання
16. Написання, компіляція і виконання програми з використанням трьох ниток виконання.

Лабораторна робота №5

Тема: Семафори в Linux як засіб синхронізації процесів

1. Семафори в UNIX.
2. Відмінність операцій над UNIX-семафорами від класичних операцій.
3. Створення масиву семафорів або доступ до вже існуючого масиву. Системний виклик `semget()`.
4. Виконання операцій над семафорами. Системний виклик `semop()`.
5. Видалення набору семафорів із системи за допомогою команди `ipcrm` або системного виклику `semctl()`.
6. Поняття про POSIX-семафори.
7. Написання, компіляція і виконання програм, що синхронізують свої дії за допомогою семафорів.
8. Написання, компіляція і виконання програми з організацією взаємної черговості за допомогою семафорів для двох процесів, що взаємодіють через `pipe`

Лабораторна робота №6

Тема: Черги повідомлень в Linux

1. Повідомлення як засоби зв'язку і засоб синхронізації процесів.
2. Черги повідомлень в UNIX як складова частина System V IPC.
3. Створення черги повідомлень або доступ до вже існуючої. Системний виклик `msgget()`.
4. Реалізація примітивів `send` і `receive`. Системні виклики `msgsnd()` і `msgrcv()`.
5. Видалення черги повідомлень із системи за допомогою команди `ipcrm` або системного виклику `msgctl()`.
6. Поняття мультиплексування. Мультиплексування повідомлень.
7. Модель взаємодії процесів клієнт-сервер. Нерівноправність клієнта і сервера.
8. Використання черг повідомлень для синхронізації роботи процесів.
9. Компіляція і виконання програми з однонапрявленою передачею текстової інформації
10. Написання, компіляція і виконання програми для передачі числової інформації
11. Написання, компіляція і виконання програм для здійснення двостороннього зв'язку через одну чергу повідомлень
12. Написання, компіляція і виконання програми сервера і клієнтів для запропонованої схеми мультиплексування повідомлень.
13. Реалізація семафорів через черги повідомлень.

Лабораторна робота №7

Тема: Організація файлової системи в Linux. Робота з файлами та директоріями. Поняття про метогу `marked` файли

1. Розділи носія інформації (`partitions`) в UNIX.
2. Логічна структура файлової системи і типи файлів в UNIX.
3. Організація файлу на диску в UNIX на прикладі файлової системи `s5fs`.
4. Поняття індексного вузла (`inode`).
5. Організація директорій (каталогів) в UNIX.
6. Поняття суперблоку.
7. Операції над файлами та директоріями.
8. Системні виклики та команди для виконання операцій над файлами і директоріями. Системний виклик `open()`. Системний виклик `close()`.
9. Операція створення файлу. Системний виклик `creat()`. Операція читання атрибутів файлу. Системні виклики `stat()`, `fstat()` і `lstat()`. Операції зміни атрибутів файлу.

10. Операції читання з файлу і запису у файл. Операція зміни покажчика поточної позиції. Системний виклик lseek(). Операція додавання інформації у файл. Прапорець O_APPEND.
11. Операції створення зв'язків. Команда ln, системні виклики link() і symlink(). Операція видалення зв'язків і файлів. Системний виклик unlink().
12. Спеціальні функції для роботи із вмістом директорій.
13. Поняття про файли, відображувані в пам'ять (memoгу marked файли). Системні виклики mmap(), munmap().
14. Написання програми, яка роздруковує вміст заданої директорії у форматі, аналогічному формату видачі команди ls -al.
15. Аналіз, компіляція і виконання програми для створення memoгу marked файлу і запису його вмісту
16. Модифікація програми для читання з файлу, використовуючи його відображення на пам'ять
17. Написання дві програми, які використовують memoгу marked файл для обміну інформацією при одночасній роботі, подібно тому, як вони могли б використати розділювану пам'ять.

Лабораторна робота №8

Тема: Організація вводу-виводу в Linux. Файли пристроїв. Апарат переривань. Сигнали в Linux

1. Поняття віртуальної файлової системи.
2. Операції над файловими системами. Монтування файлових систем.
3. Блокові, символні пристрої. Поняття драйвера. Блокові, символні драйвери, драйвери низького рівня.
4. Файловий інтерфейс.
5. Апаратні переривання (interrupt), виключення (exception), програмні переривання (trap, software interrupt). Їхня обробка.
6. Поняття сигналу. Способи виникнення сигналів і види їхньої обробки.
7. Поняття групи процесів, сеансу, лідера групи, лідера сеансу, керуючого терміналу сеансу. Системні виклики getpgrp(), setpgrp(), getpgid(), setpgid(), getsid(), setsid()
8. Системний виклик kill() і команда kill(). Системний виклик signal(). Встановлення власного обробника сигналу. Відновлення попередньої реакції на сигнал. Сигнали SIGUSR1 і SIGUSR2.
9. Використання сигналів для синхронізації процесів. Завершення породженого процесу. Системний виклик waitpid(). Сигнал SIGCHLD. Виникнення сигналу SIGPIPE при спробі запису в pipe або FIFO, який ніхто не збирається читати.
10. Поняття про надійність сигналів. POSIX-функції для роботи із сигналами.
11. Написання програм обробки сигналів.
12. Організація побітової передачі цілого числа між двома процесами, використовуючи для цього тільки сигнали SIGUSR1 і SIGUSR2.
13. Написання програми для ілюстрації обробки сигналу SIGCHLD.
14. Написати програму обробки сигналу в програмі, не застосовуючи POSIX-сигнали, так, щоб процес-батько повідомляв про статус всіх процесів-дітей, що завершилися.

Лабораторна робота №9

Тема: Сім'я протоколів TCP/IP. Сокети (sockets) в Linux і основи роботи з ними

1. Коротка історія сім'ї протоколів TCP/IP.
2. Загальні відомості про архітектуру сім'ї протоколів TCP/IP.
3. Рівень мережного інтерфейсу. Рівень Internet. Протоколи IP, ICMP, ARP, RARP.
4. Internet-адреси. Транспортний рівень. Протоколи TCP і UDP. UDP і TCP сокети (sockets).
5. Адресні простори портів. Поняття encapsulation. Рівень додатків/програм.
6. Використання моделі клієнт-сервер при вивченні мережного програмування.
7. Організація зв'язку між віддаленими процесами за допомогою датаграм. Мережний порядок байт.
8. Функції htons(), htonl(), ntohs(), ntohl(). Функції перетворення IP-адрес inet_ntoa(), inet_aton(). Функція bzero(). Системні виклики socket(), bind(), sendto(), recvfrom().
9. Організація зв'язку між процесами за допомогою встановлення логічного з'єднання. Системні виклики connect(), listen(), accept().
10. Використання інтерфейсу сокетів для інших сімей протоколів. Файли типу "сокет".
11. Написання програми UDP-клієнта та UDP-сервера.
12. Написання програми TCP-клієнта та TCP-сервера

Критерії оцінювання навчальних досягнень студентів за шкалою ECTS та національною шкалою

Форми проведення поточного контролю, їх періоди визначаються робочим планом викладача. Поточний контроль проводиться у вигляді контрольного заходу за модулем 1, модулем 2, модулем 3 заліків з лабораторних, самостійних і індивідуальних робіт. Форми підсумкового семестрового контролю визначаються навчальним планом спеціальності. Для даної спеціальності встановлено семестровий іспит по завершенню вивчення дисципліни.

Шкала оцінок контролю знань студентів

Оцінювання знань студентів виконується згідно порядку оцінювання знань студентів в умовах кредитно-модульної системи організації навчального процесу.

Критерії оцінки знань, складання іспиту

Критерії оцінки успішності повинні відповідати навчальній програмі й найбільш важливим вимогам до знань студентів:

1. Знання фактів, явищ. Вірне, науково достовірне їх пояснення.
2. Оволодіння науковими термінами, поняттями, законами, методами, правилами; вміння користуватися ними при поясненні нових фактів, розв'язуванні різних питань і виконанні практичних завдань.
3. Максимальна ясність, точність викладу думки, вміння відстоювати свої погляди, захищати їх.
4. Знання повинні мати практичну значимість: застосування їх безпосередньо на комп'ютері.

Усні відповіді повинні бути повними, логічними, доказовими. Письмові відповіді, у тому числі звіти про самостійні, індивідуальні роботи повинні бути виконані з точним дотриманням методичних вказівок, інструкцій, акуратно оформлені.

Оцінка ECTS	Критерії	За 100-бальною шкалою	За національною шкалою
A	відмінний рівень знань (умінь) в межах обов'язкового матеріалу з, можливими, незначними недоліками	90 – 100 балів	відмінно
B	достатньо високий рівень знань (умінь) в межах обов'язкового матеріалу без суттєвих (грубих) помилок	82-89 балів	добре
C	в цілому добрий рівень знань (умінь) з незначною кількістю помилок	75-81 балів	
D	посередній рівень знань (умінь) із значною кількістю недоліків, достатній для подальшого навчання або професійної діяльності	69-74 балів	задовільно
E	мінімально можливий допустимий рівень знань (умінь)	60-68 балів	
FX	незадовільний рівень знань, з можливістю повторного перескладання за умови належного самостійного доопрацювання	35-59 балів	незадовільно
F	дуже поганий рівень знань (умінь), що вимагає повторного вивчення дисципліни	0-34 балів	

Рекомендована література

1. Шеховцов В.А. Операційні системи. – К.: Видавнича група ВНУ, 2005. – 576с.: іл.
2. Столлингс В. Операционные системы. – М.: Вильямс, 2001.
3. Таненбаум Э. Современные операционные системы. – СПб.: Издательский дом Питер, 2002.
4. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб.: Издательский дом Питер, 2001.
5. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. – СПб.: Издательский дом Питер, 2003.
6. Костромин В.А. Самоучитель Linux для пользователя. – СПб.: БХВ – Петербург, 2003. – 672 с.: ил.
7. Майкл К. Джонсон, Эрик В. Троан. Разработка приложений в среде Linux. Программирование для linux. 2-е издание. – Вильямс, 2007 г. – 544 с.
8. Снейдер Й. Эффективное программирование TCP/IP. – Издательский дом Питер, 2001.
9. Таненбаум Э. Компьютерные сети. – СПб.: Издательский дом Питер, 2003.
10. Стивенс У. UNIX: разработка сетевых приложений. – СПб.: Издательский дом Питер, 2003.