

**Критерії оцінювання на екзамені
з курсу «Операційні системи»
(напрямок 040301 – прикладна математика)**

Екзаменаційний білет складається з двох теоретичних питань, кожне з яких оцінюється у десять балів, та одного практичного завдання, яке оцінюється в десять балів. Разом з результатами модульного контролю (70 балів) маємо 100 балів.

За відсутність частини відповіді на питання 1-2 знімається до 9 балів. За принципову помилку, яка приведе до втрати правильності відповіді при розв'язуванні практичного завдання, знімається 3-6 балів.

За помилки в поясненнях, назвах функцій, які не привели до принципових змін відповіді на питання, знімається до 3-х балів.

Часткова відповідь на питання, в якій немає повного розуміння понять, пояснень результатів оцінюється не вище 3-х балів у питаннях 1-2 і не вище 4-х балів при розв'язуванні задачі.

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)	50-68 балів	
FX	незадовільний рівень знань, з можливістю повторного перескладання за умови належного самостійного доопрацювання	35-49 балів	незадовільно
F	дуже поганий рівень знань (умінь), що вимагає повторного вивчення дисципліни	1-34 балів	

Питання до іспиту з курсу «Операційні системи»

1. Структура обчислювальної системи. Операційна система як віртуальна машина, як менеджер ресурсів, як захисник користувачів і програм, як постійно функціонуюче ядро.
2. Основні поняття, концепції ОС (системні виклики, переривання, виняткові ситуації, файли, процеси, нитки).
3. Архітектурні особливості ОС (монолітне ядро, багаторівневі системи (layered systems), віртуальні машини, мікроядерна архітектура, змішані системи).
4. Класифікація ОС (реалізація багатозадачності, підтримка багатокористувацького режиму, багатопроесорна обробка, системи реального часу).
5. Поняття процесу. Стани процесу.
6. 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 процесів. Критерії планування і вимоги до алгоритмів планування.
8. Параметри планування процесів. Витісняльне і невитісняльне планування.
9. Алгоритми планування процесів First-Come First-Served (FCFS) та Round Robin (RR).
10. Алгоритми планування процесів Shortest-Job-First (SJF), гарантоване планування, пріоритетне планування.
11. Алгоритми планування процесів багаторівневі черги (Multilevel Queue) та багаторівневі черги зі зворотнім зв'язком.
12. Процеси, що взаємодіють. Причини взаємодії процесів. Категорії засобів обміну інформацією.
13. 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).
14. Нитки виконання.
15. Interleaving, race condition і mutual exclusion. Активність, атомарні операції активностей, умови Бернстайна.
16. Критична секція процесу. Вимоги до алгоритмів взаємодії процесів. Програмні алгоритми організації взаємодії процесів (заборона переривань, строге чергування, прапорці готовності, алгоритм Петерсона). Алгоритм булочної організації взаємодії процесів (bakery algorithm)).
17. Механізми синхронізації процесів. Семафори. Монітори. Повідомлення. Еквівалентність семафорів, моніторів і повідомлень.
18. Поняття тупика. Умови виникнення тупиків. Основні напрямки боротьби з тупиками. Ігнорування проблеми тупиків.
19. 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).
20. Виявлення тупиків. Відновлення після тупиків.
 21. Спрощене поняття про будову файлової системи в Linux. Повні імена файлів. Поняття про поточний каталог. Команда `pwd`. Відносні імена файлів. Домашній каталог користувача.
 22. Користувач і група. Команди `chown` і `chgrp`. Права доступу до файлу. Команда `ls` з опціями `-al`. Використання команд `chmod` і `umask`.
 23. Поняття процесу, його контекст. Ідентифікація процесу. Стани процесу. Коротка діаграма станів.
 24. Ієрархія процесів в Linux. Системні виклики `getpid()`, `getppid()`.
 25. Створення процесу в Linux. Системний виклик `fork()`. Завершення процесу. Функція `exit()`.
 26. Параметри функції `main()` у мові C. Змінні середовища та аргументи командного рядка.
 27. Зміна користувацького контексту процесу. Сім'я функцій для здійснення системного виклику `exec()`.
 28. Поняття потоку вводу-виводу. Переваги та недоліки потокового обміну даними.
 29. Робота з файлами через системні виклики і стандартну бібліотеку вводу-виводу. Поняття файлового дескриптора. Відкриття файлу. Системний виклик `open()`. Системні виклики `close()`, `read()`, `write()`.
 30. Поняття `pipe`. Системний виклик `pipe()`. Організація зв'язку через `pipe` між процесом-батьком і процесом-нащадком.
 31. Успадкування файлових дескрипторів при викликах `fork()` і `exec()`. Особливості поводження викликів `read()` і `write()` для `pipe`'а.
 32. Поняття FIFO. Використання системного виклику `mknod()` для створення FIFO. Функція `mkfifo()`. Особливості поводження виклику `open()` при відкритті FIFO.
 33. Поняття System V IPC. Простір імен. Адресація в System V IPC. Функція `ftok()`. Дескриптори System V IPC. Команди `ipc` і `ipcrm`.
 34. Розділювана пам'ять в Linux. Системні виклики `shmget()`, `shmat()`, `shmdt()`. Використання системного виклику `shmctl()` для звільнення ресурсу.
 35. Розділювана пам'ять та системні виклики `fork()`, `exec()` і функція `exit()`.
 36. Поняття про нитки виконання (`thread`) в Linux. Ідентифікатор нитки виконання. Функція `pthread_self()`. Створення і завершення `thread`'а. Функції `pthread_create()`, `pthread_exit()`, `pthread_join()`. Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять.
 37. Семафори в Linux. Відмінність операцій над Linux-семафорами від класичних операцій. Створення масиву семафорів або доступ до вже існуючого масиву. Системний виклик `semget()`.
 38. Виконання операцій над семафорами. Системний виклик `semop()`. Видалення набору семафорів із системи за допомогою команди `ipcrm` або системного виклику `semctl()`.
 39. Повідомлення як засіб зв'язку і засіб синхронізації процесів. Черги повідомлень в Linux як складова частина System V IPC. Створення черги повідомлень або доступ до вже існуючої. Системний виклик `msgget()`.
 40. Реалізація для черг примітивів `send` і `receive`. Системні виклики `msgsnd()` і `msgrcv()`. Видалення черги повідомлень із системи за допомогою команди `ipcrm` або системного виклику `msgctl()`.