Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету
математики та інформатики

_____ (Мартинюк О.В.)
“12” серпня 2024 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

Операційні системи
(назва навчальної дисципліни)
Обов’язкова навчальна дисципліна
(вказати: обов’язкова / вибіркова)
Освітньо-професійна програма Технології програмування та комп’ютерне моделювання
(назва програми)
Спеціальність *113 - Прикладна* математика
(вказати: код, назва)
Галузь знань 11 - Математика та статистика

Рівень вищої освіти перший (бакалаврський)
(вказати: перший бакалаврський/другий магістерський)
Факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці – 2024

Робоча програма обов'язкової навчальної дисципліни «Операційні системи» відповідає змісту навчального курсу, складена на основі Освітньо-професійної програми “Технології програмування та комп’ютерне моделювання”.

Розробник: Данилюк Іван Михайлович, канд. фіз.-мат. наук, доцент кафедри прикладної математики та інформаційних технологій
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Погоджено з гарантом ОПП і затверджено на засіданні кафедри ПМІТ

Протокол № 1 від “8” серпня 2024 року

Завідувач кафедри _____ Бігун Я.Й.
(підпис) (прізвище та ініціали)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від “12” серпня 2024 року

Голова методичної ради факультету математики та інформатики _____ Сікора В.С.
(підпис) (прізвище та ініціали)

1. Мета та завдання навчальної дисципліни

Курс 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 сформувати у студентів уявлення про будову та принципи роботи, стан і перспективи розвитку сучасних операційних систем.

У результаті вивчення дисципліни студенти повинні

знати:

−про еволюцію обчислювальних систем,

−основні функції операційних систем і принципи їхньої побудови,

−процеси, їх стани і операції над ними,

−планування процесів,

−кооперація процесів і основні аспекти її логічної організації,

−критичні секції процесів, взаємовиключення і організація правильної черговості,

−алгоритми синхронізації процесів

−семафори, монітори, повідомлення і їх еквівалентність,

−тупики і боротьба з ними.

вміти:

−використовувати команди операційної системи Linux для керування файловою системою,

−створювати процеси в Linux, змінювати користувальницький контекст процесу за допомогою системних викликів,

−організовувати взаємодію процесів через pipe і FIFO в Linux,

−використовувати засоби System V IPC,

−працювати з розділюваною пам’яттю ОС,

−створювати нові нитки виконання в межах одного процесу,

−використовувати семафори і черги повідомлент для синхронізації процесів,

−здійснювати зв’язок між процесами з використанням розділюваної пам’яті та черги повідомлень.

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

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів компетентностей, передбачених відповідним стандартом вищої освіти України (за наявності) або ОП, програмні результати навчання якої відповідають вимогам Національної рамки кваліфікацій (за відсутності стандарту):

- загальних:

ЗК01 - здатність учитися і оволодівати сучасними знаннями;

ЗК02 - здатність застосовувати знання у практичних ситуаціях;

ЗК03 - здатність генерувати нові ідеї (креативність);

ЗК04 - здатність бути критичним і самокритичним;

ЗК06 - здатність до абстрактного мислення, аналізу та синтезу;

ЗК07 - здатність до пошуку, оброблення та аналізу інформації з різних джерел;

ЗК08 - знання та розуміння предметної області та розуміння професійної діяльності;

ЗК12 - визначеність і наполегливість щодо поставлених завдань і взятих обов'язків;

ЗК13 - навички міжособистісної взаємодії;

- фахових:

ФК04 - здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію;

ФК06 - здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків;

ФК08 - здатність використовувати сучасні технології програмування та тестування програмного забезпечення;

ФК10 - здатність створення документів встановленої звітності, використання нормативно-правових документів;

ФК17 - здатність до використання новітніх інформаційно-комунікаційних технологій.

Результати вивчення дисципліни деталізують такі програмні результати навчання ОП:

ПРН11 - вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів;

ПРН12 - розв'язувати окремі інженерні задачі та/або задачі, що виникають принаймні в одній предметній галузі: в соціології, економіці, екології та медицині;

ПРН14 - виявляти здатність до самонавчання та продовження професійного розвитку;

ПРН15 - уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу;

ПРН18 - ефективно спілкуватися з питань інформації, ідей, проблем та рішень зі спеціалістами та суспільством загалом

ПРН22 - знати архітектуру сучасних обчислювальних систем та комп'ютерних мереж, основні мережні технології; вміти проектувати, адмініструвати та здійснювати захист комп'ютерних мереж.

2. Теоретичний зміст програми навчальної дисципліни

Змістовий модуль 1. Поняття операційної системи. Процеси. Базові аспекти організації, планування та взаємодії процесів.

Тема 1. 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 багатокористувацького режиму, багатопроцесорна обробка, системи реального часу).

Тема 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).

Тема 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)).

Тема 4. Кооперація процесів і основні аспекти її логічної організації.

Процеси, що взаємодіють. Категорії засобів обміну інформацією. Логічна організація механізму передачі информації (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.

Тема 5. Знайомство з операційною системою Linux.

Вступ до курсу практичних занять. Коротка історія операційної системи Linux, її структура. Системні виклики та бібліотека libc. Поняття login і password. Спрощене поняття про будову файлової системи в Linux. Повні імена файлів. Поняття про поточний каталог. Команда pwd. Відносні імена файлів. Домашній каталог користувача. Команда man – універсальний довідник. Команди cd – зміни поточного каталогу і ls – перегляду вмісту каталогу. Команда cat і створення файлу. Перенапрямлення вводу/виводу. Найпростіші команди для роботи з файлами – cp, rm, 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 Linux і запуск їх на виконання.

Тема 6. Процеси в операційній системі Linux.

Поняття процесу в Linux, його контекст. Ідентифікація процесу. Стан процесу. Коротка діаграма станів. Ієрархія процесів. Системні виклики getpid(), getppid(). Створення процесу в Linux. 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().

Тема 7. Організація взаємодії процесів через pipe і FIFO в Linux.

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.

Змістовий модуль 2. Розширені аспекти організації, планування та взаємодії процесів.

Тема 8. 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)).

Тема 9. 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).

Тема 10. 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.

Тема 11. Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).

Переваги та недоліки потокового обміну даними. Поняття System V IPC. Простір імен. Адресація в System V IPC. Функція ftok(). Дескриптори System V IPC. Розділювана пам'ять в Linux. 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 Linux. 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.

Тема 12. Семафори в Linux як засіб синхронізації процесів.

Семафори в Linux. Відмінність операцій над Linux-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.

Тема 13. Черги повідомлень в Linux.

Повідомлення як засоби зв'язку і засоб синхронізації процесів. Черги повідомлень в Linux 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.

3. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Поняття операційної системи. Процеси. Базові аспекти організації, планування та взаємодії процесів.												
Тема 1. Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.	4	2				2						
Тема 2. Процеси. Їх стани і операції над ними.	8	4				4						
Тема 3. Планування процесів.	7	4				3						
Тема 4. Кооперація процесів і основні аспекти її логічної організації.	5	2				3						
Тема 5. Знайомство з операційною системою Linux.	15	1		4		10						
Тема 6. Процеси в операційній системі Linux.	15	2		5		8						
Тема 7. Організація взаємодії процесів через ріре і FIFO в Linux.	18	1		5		12						
Разом за змістовим модулем 1	72	16	0	14		42						
Змістовий модуль 2. Розширені аспекти організації, планування та взаємодії процесів.												
Тема 8. Критичні секції процесів, взаємовиключення і організація правильної черговості. Алгоритми синхронізації процесів.	8	4				4						
Тема 9. Семафори, монітори, повідомлення і їх еквівалентність.	8	4				4						
Тема 10. Тупики і боротьба з ними.	6	2				4						
Тема 11. Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).	22	2		6		14						
Тема 12. Семафори в Linux як засіб синхронізації процесів.	17	1		4		12						
Тема 13. Черги повідомлень в Linux.	17	1		6		10						
Разом за змістовим модулем 2	78	14	0	16		48						
Усього годин	150	30	0	30		90						

4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Знайомство з ОС Linux.	4
2	Процеси в ОС Linux.	5
3	Організація взаємодії процесів через pipe і FIFO в Linux.	5
4	Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).	6
5	Семафори в Linux як засіб синхронізації процесів.	4
6	Черги повідомлень в Linux.	6
	Разом	30

5. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.	2
2	Процеси. Їх стани і операції над ними.	4
3	Планування процесів.	3
4	Кооперація процесів і основні аспекти її логічної організації.	3
5	Знайомство з операційною системою Linux.	10
6	Процеси в операційній системі Linux.	8
7	Організація взаємодії процесів через pipe і FIFO в Linux.	12
8	Критичні секції процесів, взаємовиключення і організація правильної черговості. Алгоритми синхронізації процесів.	4
9	Семафори, монітори, повідомлення і їх еквівалентність.	4
10	Тупики і боротьба з ними.	4
11	Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).	14
12	Семафори в Linux як засіб синхронізації процесів.	12
13	Черги повідомлень в Linux.	10
	Разом	90

6. Методи навчання

Під час вивчення курсу використовуються словесні методи навчання (розповідь, діалог), метод презентацій, демонстрації. Проте основне навчання відбувається за допомогою виконання лабораторних робіт.

7. Методи контролю

Форми проведення поточного контролю, їх періоди визначаються робочим планом викладача. Поточний контроль проводиться у вигляді двох контрольних робіт, заліків з лабораторних робіт. Форми підсумкового семестрового контролю визначаються навчальним планом спеціальності. Для даної спеціальності встановлено семестровий іспит по завершенню вивчення дисципліни.

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

8. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота								Іспит	Сума
Змістовий модуль №1				Змістовий модуль № 2					
ЛБ1	ЛБ2	ЛБ3	КР1	ЛБ4	ЛБ5	ЛБ6	КР2	30	100
5	10	10	10	9	7	9	10		

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	**A**	відмінно	зараховано
80-89	**B**	добре	
70-79	**C**		
60-69	**D**	задовільно	
50-59	**E**		
35-49	**FX**	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	**F**	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

9. Питання до іспиту

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, багатопроцесорна 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 С. 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 pip'a.
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'a. 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().

10. Рекомендована література

1. Електронні матеріали лекцій та завдання на лабораторні роботи

   http://os.fast-page.org/?i=1

2. Електронний курс

   https://moodle.chnu.edu.ua/course/view.php?id=3449

3. Данилюк І.М. Операційні системи. Практикум: навчальний посібник / І.М. Данилюк. – Чернівці: Чернівецький нац. ун-т, 2015. – 207 с.”
4. Шеховцов В.А. Операційні системи. – К.: Видавнича група BHV, 2005. – 576с.: іл.
5. William Stallings. Operating Systems: Internals and Design Principles. - Pearson; 8th edition, 2014. - 800 p.
6. Andrew S. Tanenbaum, Herbert Bos. Modern operating systems. 4th edition. - Pearson, 2015. - 1137 p.
   https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20-%20Modern%20Operating%20Systems.pdf
7. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. Operating system concepts. - John Wiley & Sons, 2013. - 944 p.
   https://drive.uqu.edu.sa/_/mskhayat/files/MySubjects/2017SS%20Operating%20Systems/Abraham%20Silberschatz-Operating%20System%20Concepts%20(9th,2012_12).pdf