

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Краснодонский факультет инженерии и менеджмента (филиал)  
Кафедра информационных технологий и транспорта



УТВЕРЖДАЮ:

Директор

Панайотов К.К.

(подпись)

«14» марта 2025 года

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по учебной дисциплине

**Системное программирование**

(наименование учебной дисциплины, практики)

**09.03.01 Информатика и вычислительная техника**

(код и наименование направления подготовки (специальности))

**«Компьютерные системы и сети»**

наименование профиля подготовки (специальности, магистерской программы); при отсутствии ставится прочерк)

Разработчик(разработчики):

ст. преподаватель

(подпись)

Крупича О.В.

ФОС рассмотрен и одобрен на заседании кафедры информационных технологий и транспорта от «26» февраля 2025 г., протокол № 7

Заведующий кафедрой

информационных

технологий и транспорта

(подпись)

Верительник Е.А

Краснодон 2025

**Комплект оценочных материалов по дисциплине  
«Системное программирование»**

**Задания закрытого типа**

**Задания закрытого типа на выбор правильного ответа**

*Выберите один правильный ответ.*

1. Какой системный вызов в UNIX/Linux используется для создания нового процесса?

- А) fork().
- Б) exec().
- В) thread().
- Г) clone().

Правильный ответ: А

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. Какой механизм позволяет процессу работать с виртуальной памятью, превышающей размер физической памяти?

- А) Сегментация.
- Б) Страничная организация памяти.
- В) Подкачка (swapping).
- Г) Кэширование.

Правильный ответ: В

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. Какой системный вызов в Linux используется для синхронного чтения данных из файла?

- А) read().
- Б) fread().
- В) pread().
- Г) readfile().

Правильный ответ: А

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. Какой механизм IPC (Inter-Process Communication) НЕ использует ядро ОС для передачи данных?

- А) Очереди сообщений.
- Б) Разделяемая память.
- В) Сокеты.
- Г) Сигналы.

Правильный ответ: В

Компетенции (индикаторы): ПК-2 (ПК-2.2)

## Задания закрытого типа на установление соответствия

*Установите правильное соответствие.*

*Каждому элементу левого столбца соответствует только один элемент правого столбца.*

1. Установите соответствие между системными вызовами и их назначением:

Системный вызов	Назначение
1) fork()	А) Создание нового процесса.
2) exec()	Б) Замена образа процесса.
3) wait()	В) Ожидание завершения дочернего процесса.
4) exit()	Г) Завершение процесса.

Правильный ответ: 1А, 2Б, 3В, 4Г

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. Установите соответствие между понятиями и их определениями:

Понятие	Определение
1) Страничная память	А) Механизм преобразования виртуальных адресов в физические.
2) TLB	Б) Буфер ассоциативной трансляции адресов.
3) Swapping	В) Выгрузка страниц на диск.
4) MMU	Г) Устройство управления памятью.

Правильный ответ: 1А, 2Б, 3В, 4Г

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. Установите соответствие между системными вызовами и их функциями:

Системный вызов	Функция
1) open()	А) Открытие файла.
2) read()	Б) Чтение данных.
3) lseek()	В) Перемещение указателя позиции.
4) fsync()	Г) Синхронная запись данных на диск.

Правильный ответ: 1А, 2Б, 3В, 4Г

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. Установите соответствие между механизмами ИРС и их описанием:

Механизм	Описание
1) Очереди сообщений	А) Буферизованный обмен структурированными сообщениями.

- |                       |  |
|-----------------------|--|
| 2) Разделяемая память | Б) Область памяти, доступная нескольким процессам. |
| 3) Семафоры           | В) Синхронизация доступа к ресурсам.               |
| 4) Сигналы            | Г) Асинхронные уведомления между процессами.       |

Правильный ответ: 1 А, 2Б, 3В, 4Г

Компетенции (индикаторы): ПК-2 (ПК-2.2)

**Задания закрытого типа на установление правильной последовательности**

*Установите правильную последовательность.*

*Запишите правильную последовательность букв слева направо.*

1. Установите правильную последовательность этапов жизненного цикла процесса в ОС:

- А) Исполнение.
- Б) Создание.
- В) Ожидание.
- Г) Завершение.

Правильный ответ: Б, А, В, Г

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. Установите правильную последовательность этапов обработки системного вызова в Linux:

- А) Возврат в пользовательское пространство.
- Б) Переключение в режим ядра.
- В) Выполнение обработчика ядра.
- Г) Идентификация вызова.

Правильный ответ: Б, Г, В, А

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. Установите правильную последовательность действий при обработке page fault:

- А) Поиск свободного кадра.
- Б) Загрузка страницы с диска.
- В) Обновление таблицы страниц.
- Г) Генерация исключения.

Правильный ответ: Г, А, Б, В

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. Установите правильную последовательность операций при открытии файла в UNIX:

- А) Проверка прав доступа.
- Б) Поиск файла в файловой системе.

В) Создание файлового дескриптора.

Г) Чтение inode.

Правильный ответ: Б, Г, А, В

Компетенции (индикаторы): ПК-2 (ПК-2.2)

### **Задания открытого типа**

#### **Задания открытого типа на дополнение**

*Напишите пропущенное слово (словосочетание).*

1. \_\_\_\_\_ – это системный вызов в UNIX-подобных ОС, который создает точную копию текущего процесса, включая его память и контекст выполнения.

Правильный ответ: fork().

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. \_\_\_\_\_ – это аппаратно-программный механизм, который преобразует виртуальные адреса в физические и контролирует права доступа к памяти.

Правильный ответ: MMU (Memory Management Unit).

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. \_\_\_\_\_ – это структура данных в UNIX-подобных системах, содержащая метаданные файла (права доступа, владельца, размер) и указатели на блоки данных.

Правильный ответ: inode.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. \_\_\_\_\_ – это механизм ИРС, позволяющий процессам обмениваться данными через общую область памяти без копирования.

Правильный ответ: Разделяемая память (shared memory).

Компетенции (индикаторы): ПК-2 (ПК-2.2)

#### **Задания открытого типа с кратким свободным ответом**

*Дайте ответ на вопрос.*

1. Назовите как минимум три различия между процессами и потоками в операционной системе.

Правильные ответы:

Процессы изолированы друг от друга, а потоки разделяют память внутри процесса.

Создание процесса требует больше ресурсов, чем создание потока.

Переключение между потоками происходит быстрее, чем между процессами.

Каждый процесс имеет отдельное адресное пространство, а потоки используют общее.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. Перечислите три примера системных вызовов в UNIX/Linux, связанных с работой с файлами.

Правильные ответы:

``open()`` – открытие файла.

``read()`` – чтение из файла.

``write()`` – запись в файл.

``close()`` – закрытие файла.

``stat()`` – получение информации о файле.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. Назовите три механизма межпроцессного взаимодействия (IPC) в операционных системах.

Правильные ответы:

1. Каналы (Pipes) – однонаправленный обмен данными.

2. Очереди сообщений (Message Queues) – обмен структурированными сообщениями.

3. Разделяемая память (Shared Memory) – общая область памяти для процессов.

4. Семафоры (Semaphores) – синхронизация доступа к ресурсам.

5. Сокеты (Sockets) – обмен данными между процессами, в т. ч. по сети.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. Какие три стратегии управления памятью используются в современных ОС?

Правильные ответы:

Сегментация – разделение памяти на логические сегменты.

Страничная организация (Paging) – разбиение памяти на фиксированные страницы.

Виртуальная память – использование дискового пространства как расширения ОЗУ.

Свопинг (Swapping) – перемещение процессов между ОЗУ и диском.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

### **Задания открытого типа с развернутым ответом**

*Дайте развернутый ответ на вопрос.*

1. Опишите жизненный цикл процесса в операционной системе. Какие состояния может принимать процесс, и какие системные вызовы (на примере UNIX/Linux) влияют на переход между этими состояниями?

Время выполнения – 20 мин.

Ожидаемый результат:

Жизненный цикл процесса включает несколько состояний:

1. Создание (New) – процесс создается через `'fork()'` (копирование родительского процесса) или `'exec()'` (загрузка новой программы).
2. Готовность (Ready) – процесс ожидает выделения CPU.
3. Выполнение (Running) – процесс выполняется на процессоре.
4. Ожидание (Waiting/Blocked) – процесс приостановлен (например, из-за ожидания I/O или сигнала).
5. Завершение (Terminated) – процесс завершается через `'exit()'`.

Примеры системных вызовов:

- `'fork()'` – создает новый процесс-потомок.
- `'wait()'` – родительский процесс ожидает завершения потомка.
- `'kill()'` – отправка сигнала для изменения состояния (например, завершения).

Критерии оценивания: Ответ должен содержательно соответствовать ожидаемому результату.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

2. Объясните, как работает системный вызов `'read()'` в UNIX/Linux. Какие параметры он принимает, какие ошибки может возвращать и как взаимодействует с ядром ОС?

Время выполнения – 20 мин.

Ожидаемый результат:

Системный вызов `'read()'` используется для чтения данных из файлового дескриптора.

Параметры:

1. `'int fd'` – дескриптор файла (например, открытого через `'open()'`).
2. `'void *buf'` – буфер для записи данных.
3. `'size_t count'` – максимальное количество байт для чтения.

Возвращаемые значения:

- При успехе – количество прочитанных байт.
- `'0'` – достигнут конец файла.
- `'-1'` – ошибка (код ошибки в `'errno'`), например:
- `'EBADF'` – неверный дескриптор,
- `'EINTR'` – вызов прерван сигналом.

Взаимодействие с ядром:

1. Процесс переходит в режим ядра (через прерывание, например, `'int 0x80'` в x86).
2. Ядро проверяет права доступа и читает данные из кэша/диска.
3. Данные копируются в пользовательский буфер, процесс возвращается в пользовательский режим.

Критерии оценивания: Ответ должен содержательно соответствовать ожидаемому результату.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

3. Опишите жизненный цикл процесса в операционной системе. Какие состояния может принимать процесс, и какие системные вызовы (или события) приводят к переходам между этими состояниями?

Время выполнения – 20 мин.

Ожидаемый результат:

Жизненный цикл процесса включает несколько состояний:

1. Создание (New) – процесс создается системным вызовом (`fork()`, `exec()`, `CreateProcess()`).

2. Готовность (Ready) – процесс загружен в память и ожидает выделения CPU.  
- Переход в это состояние происходит после создания или при освобождении ресурсов (`sched_yield()`, завершение кванта времени).

3. Исполнение (Running) – процесс выполняется на CPU.

- Переход из Ready в Running происходит при выборе планировщиком (`scheduler`).

4. Ожидание (Blocked/Waiting) – процесс приостановлен, ожидает события (ввод-вывод, сигнал, мьютекс).

- Переход происходит при вызове `sleep()`, `wait()`, `read()` (если данные не готовы).

5. Завершение (Terminated) – процесс завершил выполнение (нормально или аварийно).

- Переход происходит после `exit()`, `kill()`, или при ошибке (segmentation fault).

Пример переходов:

- Running → Blocked: процесс вызывает `read()` из файла, который еще не загружен в память.

- Blocked → Ready: данные становятся доступны, ОС перемещает процесс в очередь готовых.

- Ready → Running: планировщик выбирает процесс для выполнения.

Критерии оценивания: Ответ должен содержательно соответствовать ожидаемому результату.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

4. Опишите механизм страничной организации памяти (Paging). Как ОС обрабатывает page faults, и какие структуры данных для этого используются?

Время выполнения – 20 мин.

Ожидаемый результат:

Принцип работы Paging:

1. Виртуальная память делится на страницы, физическая – на фреймы (обычно по 4 КБ).

2. Таблица страниц (Page Table) хранит соответствие виртуальных адресов физическим.

Page Fault (Обработка отсутствия страницы):

1. CPU генерирует исключение при обращении к странице, которой нет в ОЗУ.



2. ОС ищет страницу в файле подкачки (swap) или загружает с диска (для mmap'd файлов).

3. Если память переполнена, алгоритм замены (например, LRU) выгружает старую страницу.

Структуры данных:

- TLB (Translation Lookaside Buffer) – кэш для быстрого доступа к таблице страниц.

- Page Table Entry (PTE) – запись в таблице страниц (флаги присутствия, прав доступа).

Пример: При работе с большими массивами данных часть страниц может выгружаться в swap, что замедляет выполнение программы.

Критерии оценивания: Ответ должен содержательно соответствовать ожидаемому результату.

Компетенции (индикаторы): ПК-2 (ПК-2.2)

## Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Системное программирование» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению 09.03.01 Информатика и вычислительная техника.

Председатель учебно-методической  
комиссии Краснодарского факультета  
инженерии и менеджмента (филиала)

 Родионова О.Ю.

### Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобренны изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)