II. Управление ресурсами в OS Unix

Основные понятия ОС UNIX

Процесс – единица работы, управления и потребления ресурсов.
– программа, выполняемая в своем виртуальном адресном пространстве.
– последовательность операций программы на этапе ее выполнения.

Программа = исполняемый файл.

Ядро (Kernel) → секция управляющих структур
→ программная секция → машинно-независимая часть;
→ машинно-зависимая часть

Функции ядра:

- инициализация системы;
- выполнение процессов;
- диспетчеризация процессов;
- выделение выполняющемуся процессу оперативной памяти, управлением своппингом;
- выделение внешней памяти;
- обеспечение доступа процессов к периферийным устройствам;
- прием от процессов запросов на обслуживание

Пользователи ← **Программы-утилиты** → **Ядро**

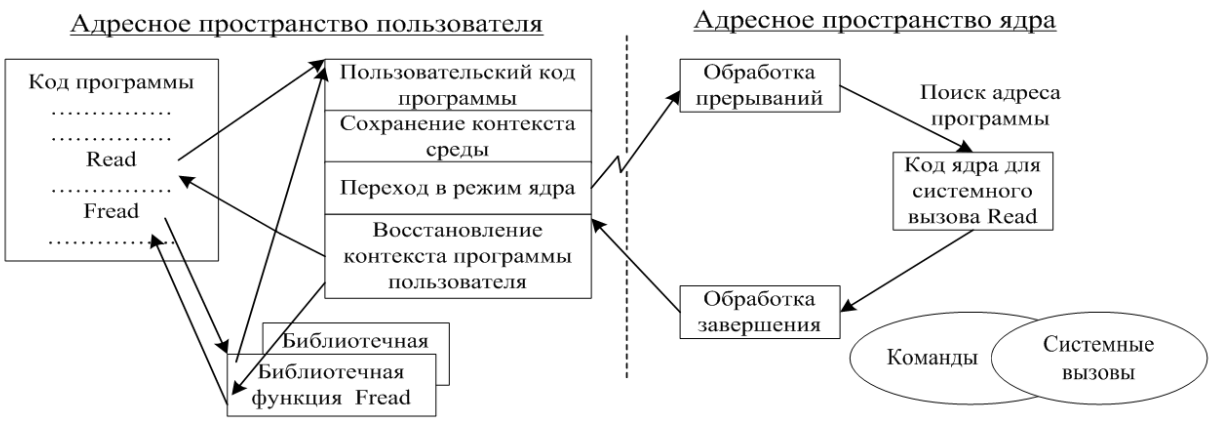
Основные понятия ОС UNIX

Два уровня выполнение пользовательских процессов в ОС Unix : уровень пользователя
уровень ядра

Пример: обработка Shell строки символов

Аппарат системных вызовов

nread = fread(inputbuf, OBJSIZE, num, fileptr);
nread = read(filefd, inputbuf, BUFSIZE);



Основные понятия ОС UNIX

Критическими секции - участки системных фаз, в которые процесс не должен входить, пока некоторый процесс не выйдет из этого участка

Аппарат событий – механизм, используемый для синхронизации процессов в системной фазе

Система приоритетов прерываний

Машинные сбои	более высокий
Системные часы	↕
Диск	
Сетевое оборудование	приоритет
Терминал	
Программные прерывания	более низкий

Механизмы режима квантования времени в ОС Unix:

- системные часы;
- системные вызовы

Свопинг – процедура перемещения образа процесса из оперативной памяти во внешнюю и обратно

Функция системы мультипрограммирования в ОС Unix:

- управление последовательностью выполнения процессов;
- управление последовательностью свопинга.

Факторы, определяющие дисциплину мультипрограммирования:

- время нахождения процесса в активном состоянии в оперативной памяти;
- время нахождения процесса во внешней памяти;
- размер оперативной памяти, занимаемый процессом, находящимся в системной фазе в режиме ожидания;
- время нахождения процесса в оперативной памяти.

Механизмы управления памятью

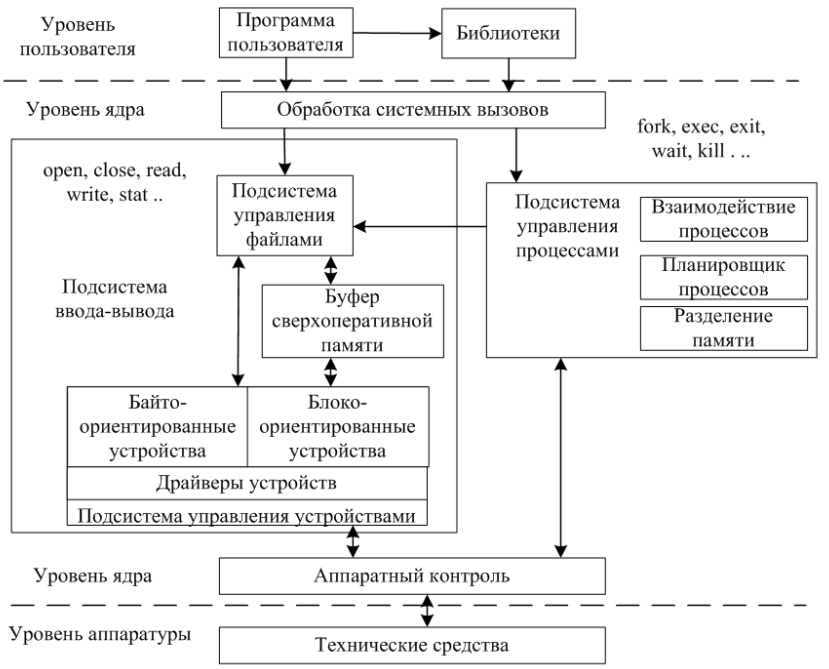
- виртуальное адресное пространство;
- выделение машинных команд в отдельный процедурный сегмент;
- использование командного языка для организации конвейера команд;
- исключение больших массивов данных из виртуального адресного пространства.

Архитектура ОС Unix

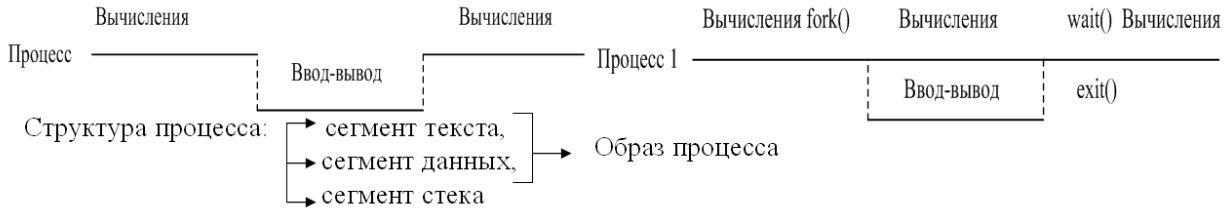
- буферизация ввода-вывода;
- упрощение структуры файла

Компоненты ядра ОС Unix:

- подсистема управления файлами (ПУФ);
- подсистема управления процессами (ПУП);
- подсистема управления устройствами (ПУУ)



Процесс



Структура процесса: сегмент текста, сегмент данных, сегмент стека → Образ процесса

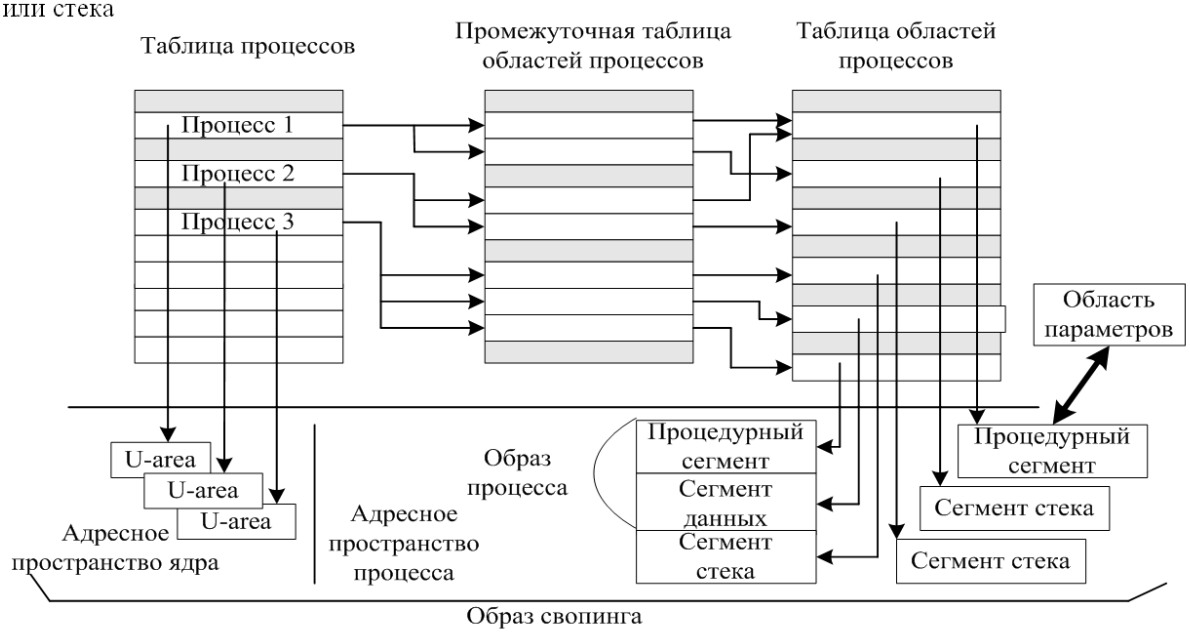
```
#include <fcntl.h>
char buffer[2048];        ← bss-данные
main (argc,argv)
    int argc;char *argv[];
  {
     int fdold, fdnew;    .
 . . . . . .
     fdopen=open(argv[1],O_RDONLY);
/* исходный файл открыт для чтения */
     fdnew=creat(argv[2],  0666);
/* создан новый файл с разрешением
чтения и записи для всех */
     copy (fdold, fdnew);
     exit(0);
  }
copy(old, new)
  int old, new;
  {    int count;
       while ((count = read(old,
                  buffer,sizeof(buffer))) > 0
            write (new, buffer, count);
}
```

	Стек задачи		Стек ядра
Запись3 Call Write()	Локальные переменные (не показаны)	Направление увеличения стека	
	Адрес возврата после вызова Write		
	Параметры, передаваемые Write		
Запись2 Call Copy()	Локальные переменные (count)	Запись2 Call func2()	Локальные переменные
	Адрес возврата после вызова Copy		Адрес возврата после вызова func2
	Параметры, передаваемые Copy (old, new)		Параметры, передаваемые ф-и ядра func2
Запись1 Call main()	Локальные переменные (fdold, fdnew)	Запись1 Call func1()	Локальные переменные
	Адрес возврата после вызова Main		Адрес возврата после вызова func1
	Параметры, передаваемые Main (argc, argv)		Параметры, передаваемые ф-и ядра func1

Процесс и его контекст

Процесс → Таблица процессов ядра → Промежуточная таблица областей процессов
→ Таблица областей процессов
U-область

Область - непрерывная зона адресного пространства, выделяемая для размещения текста, данных или стека



Контекст процесса – состояние процесса, определяемое текстом, значениями глобальных переменных, значениями используемых машинных регистров, значениями, хранимыми в таблице процессов и связанными с ней информация из U-области и таблиц областей, а также содержимым стеков и ядра, относящихся к этому процессу

Процесс и его контекст

Контекст
- пользовательский контекст
- регистровый процесс
- контекст системного уровня
  - статическая часть
  - переменное число динамических частей

Пользовательский контекст - содержимое сегментов программного кода, данных стека, разделяемых сегментов.

Регистровый контекст - содержимое аппаратных регистров, связанных с процессом

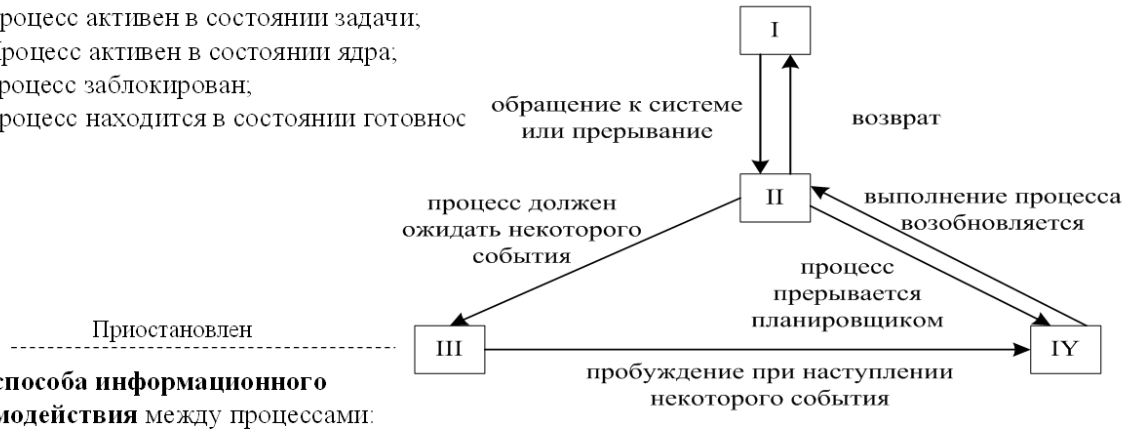
Статическая часть контекста системного уровня

- **Элемент таблицы процессов:**
  - состояние процесса;
  - физический адрес в основной или внешней памяти U-области процесса;
  - идентификатор пользователя, от имени которого запущен процесс;
  - идентификатор процесса;
  - дескриптор события, когда процесс приостановлен;
  - ссылка на промежуточную таблицу областей процессов
- **U-область (*U-area*)**:
  - указатель на элемент таблицы процессов;
  - счетчик времени, в течение которого процесс реально занимал процессор в режиме пользователя и ядра;
  - параметры и результаты системного вызова;
  - таблица дескрипторов открытых файлов;
  - предельные размеры адресного пространства;
  - параметры текущей системной операции;
  - коды ошибок

Динамическая часть контекста системного уровня – это один или несколько стеков, которые используются процессом при его выполнении в режиме ядра

Состояния и взаимодействие процессов

I. Процесс активен в состоянии задачи;
II. Процесс активен в состоянии ядра;
III. Процесс заблокирован;
IY. Процесс находится в состоянии готовнос



Три способа информационного взаимодействия между процессами:

- путем передачи внешних параметров;
- через файловую систему;
- через программный канал.