Процессы

- *Понятие процесса*
- *Состояния процесса*
- *Блок управления процессом*
- *Диспетчеризация процессов (scheduling)*
- *Операции над процессами*

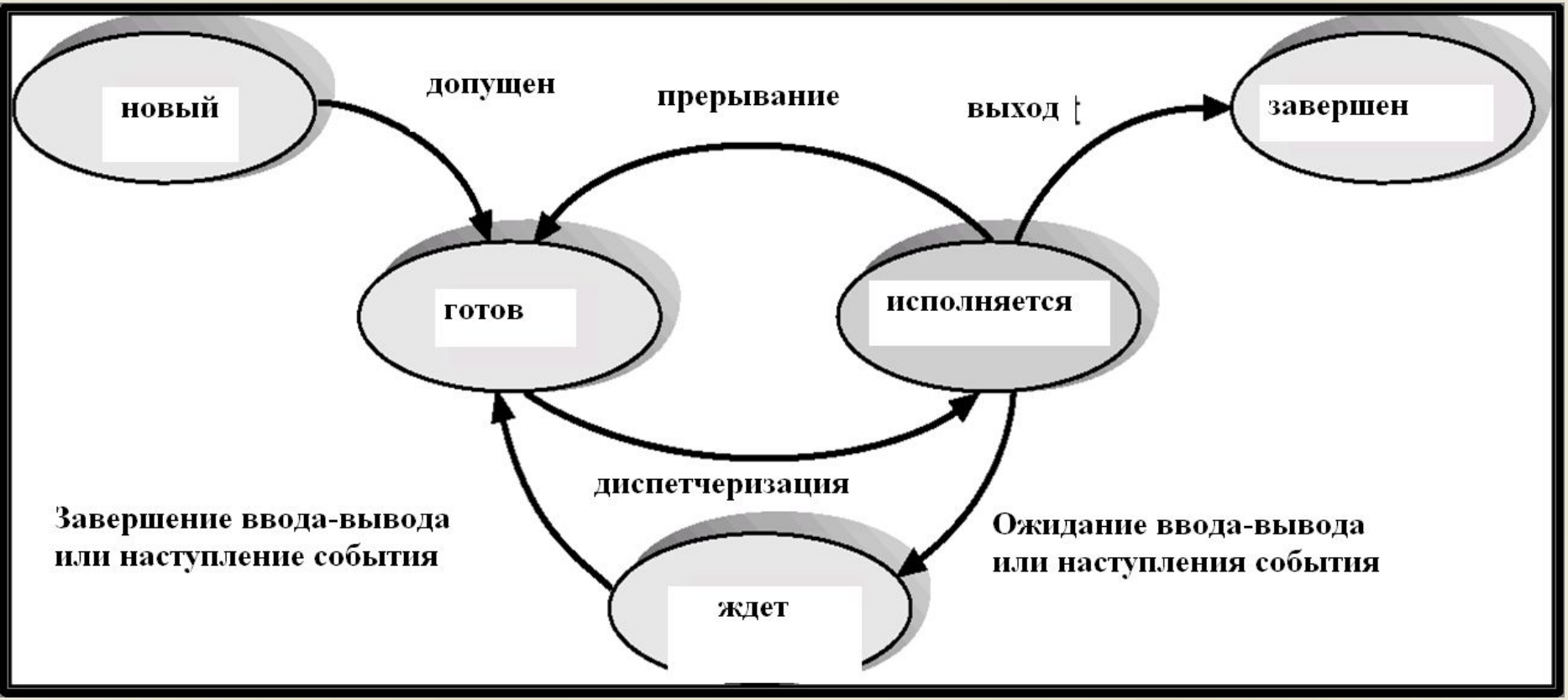
Понятие процесса

- ОС исполняет множество классов программ:
  - Пакетная система (batch system) – задания (jobs)
  - Система с разделением времени – пользовательские программы (задачи – tasks)
- Во многих учебниках термины “задание” и “процесс” – почти синонимы
- Процесс – программа при ее выполнении; он должен выполняться последовательно
- Процесс включает:
  - Счетчик команд (program counter)
  - Стек (stack)
  - Секцию данных (data section)

Состояния процесса

- При исполнении процесс может изменять свое состояние следующим образом:
  - *Новый* (new): Процесс создается.
  - *Исполняемый* (running): Исполняются команды процесса
  - *Ожидающий* (waiting): Процесс ожидает наступления некоторого события (event)
  - *Готовый к выполнению* (ready): Процесс ожидает получения ресурсов процессора для его исполнения
  - *Завершенный* (terminated): Исполнение процесса завершено.

Диаграмма состояний процесса

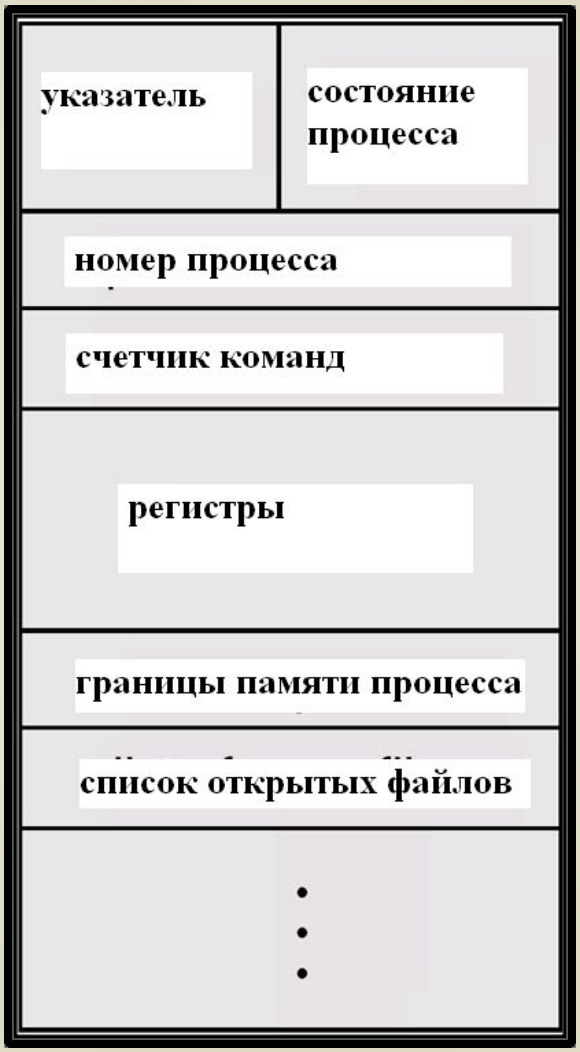


Блок управления процессом (Process Control Block – PCB)

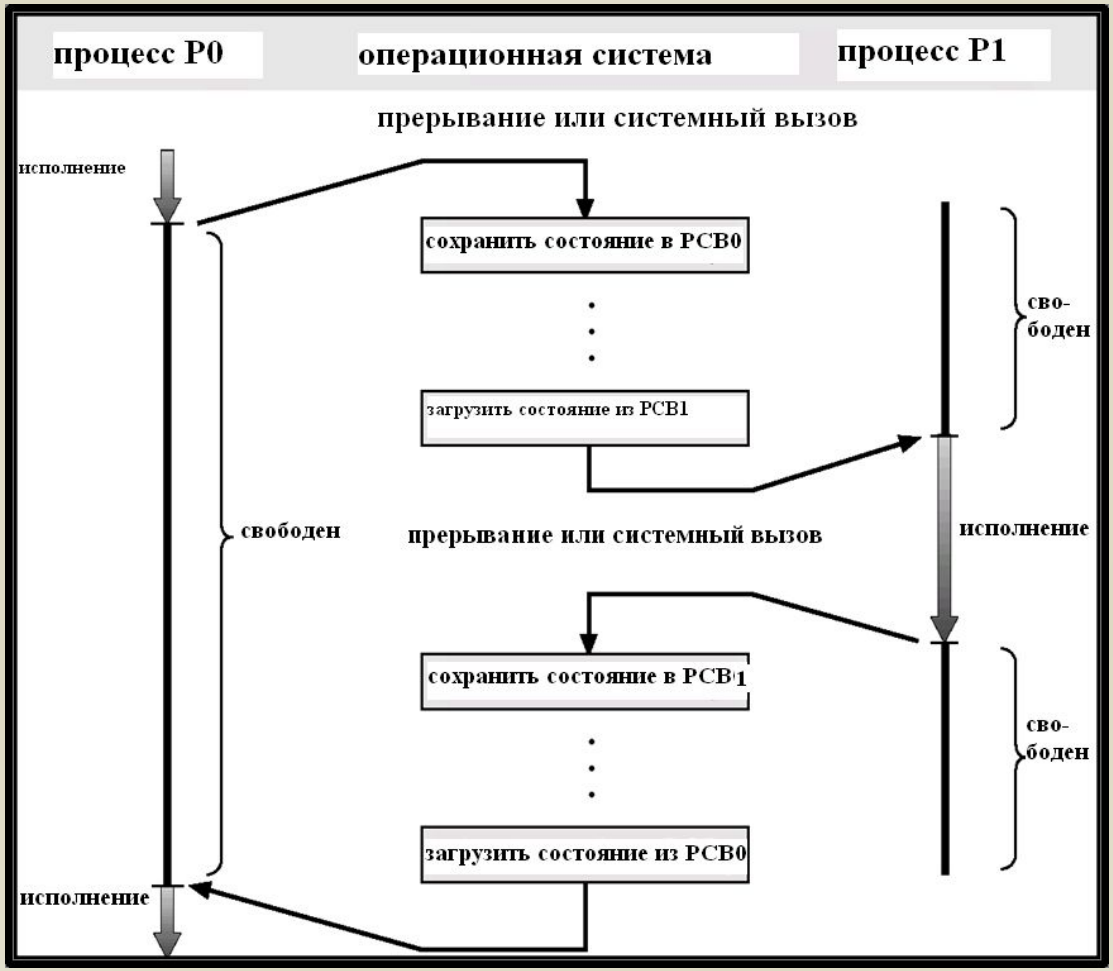
Информация, ассоциируемая с каждым процессом

- Состояние процесса
- Счетчик команд
- Регистры процессора
- Информация для диспетчеризации процессора
- Информация для управления памятью
- Статистическая информация
- Информация о состоянии ввода-вывода

Блок управления процессом (PCB)



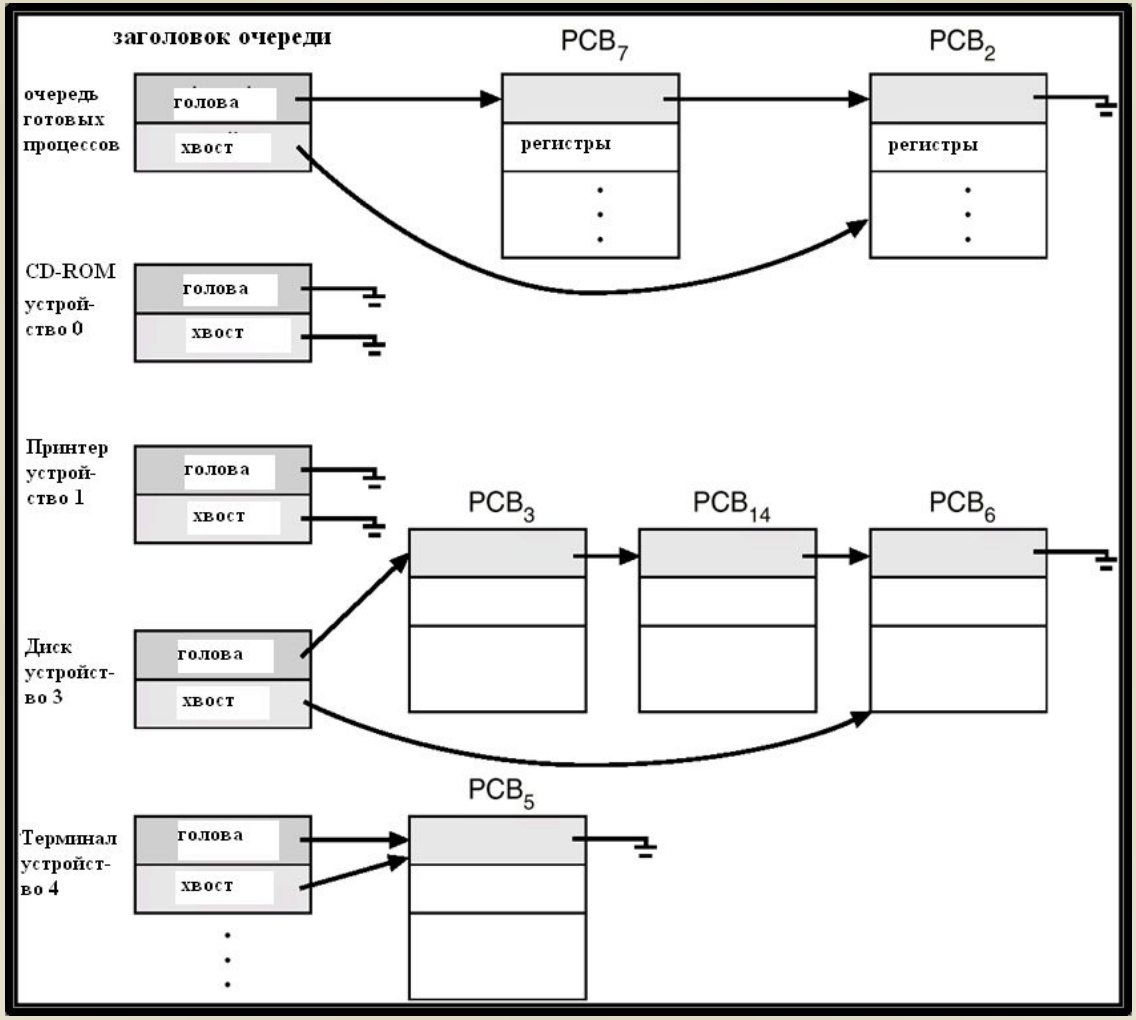
Переключение процессора с одного процесса на другой



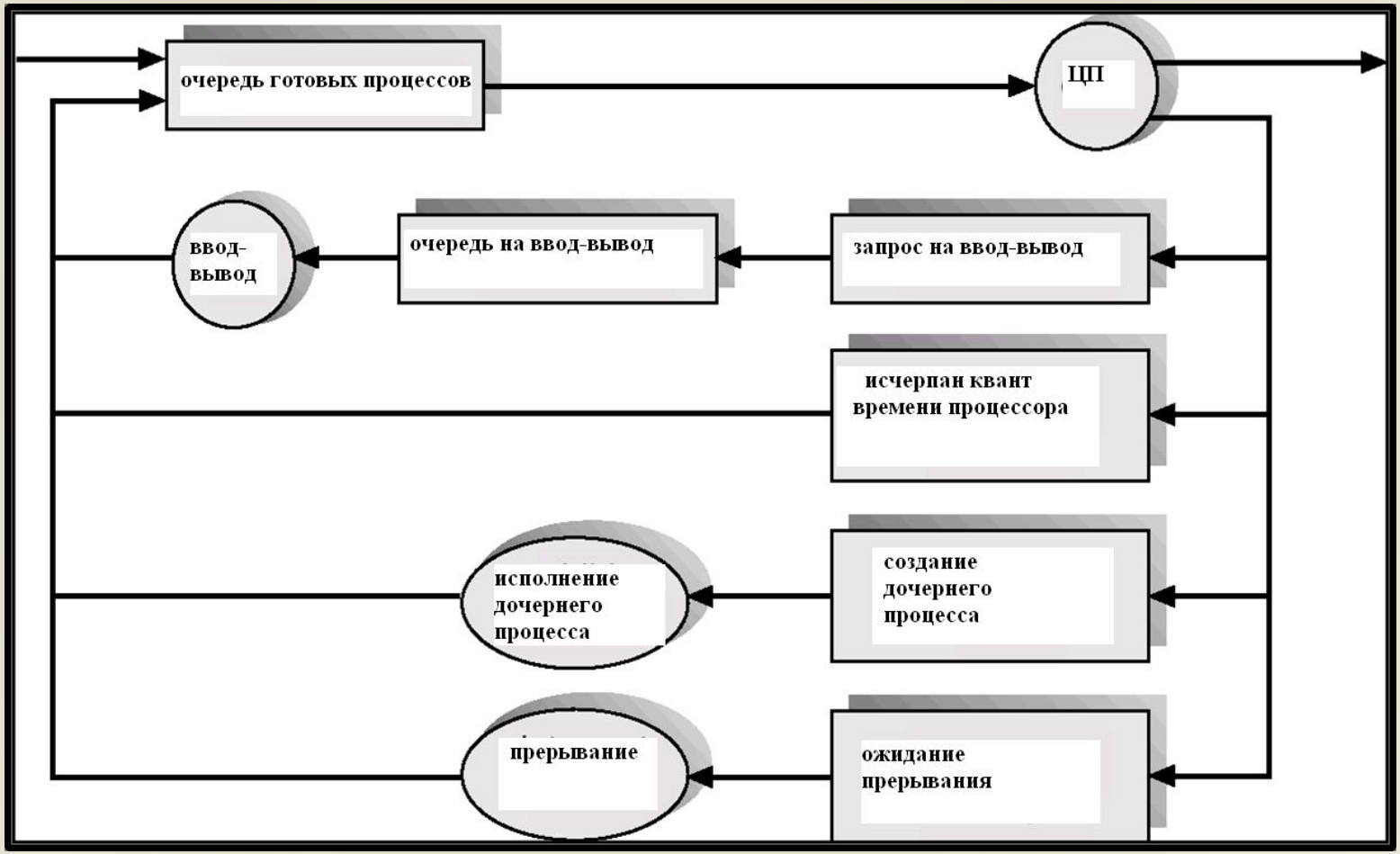
Очереди, связанные с диспетчеризацией процессов

- Очередь заданий (Job queue) – множество всех процессов в системе
- Очередь готовых процессов (Ready queue) – множество всех процессов, находящихся в основной памяти и готовых к выполнению
- Очередь ожидающих ввода-вывода (Device queues) – множество процессов, ожидающих результата работы устройства ввода-вывода
- Процессы мигрируют между различными очередями

Очередь готовых процессов и очереди к различным устройствам ввода-вывода



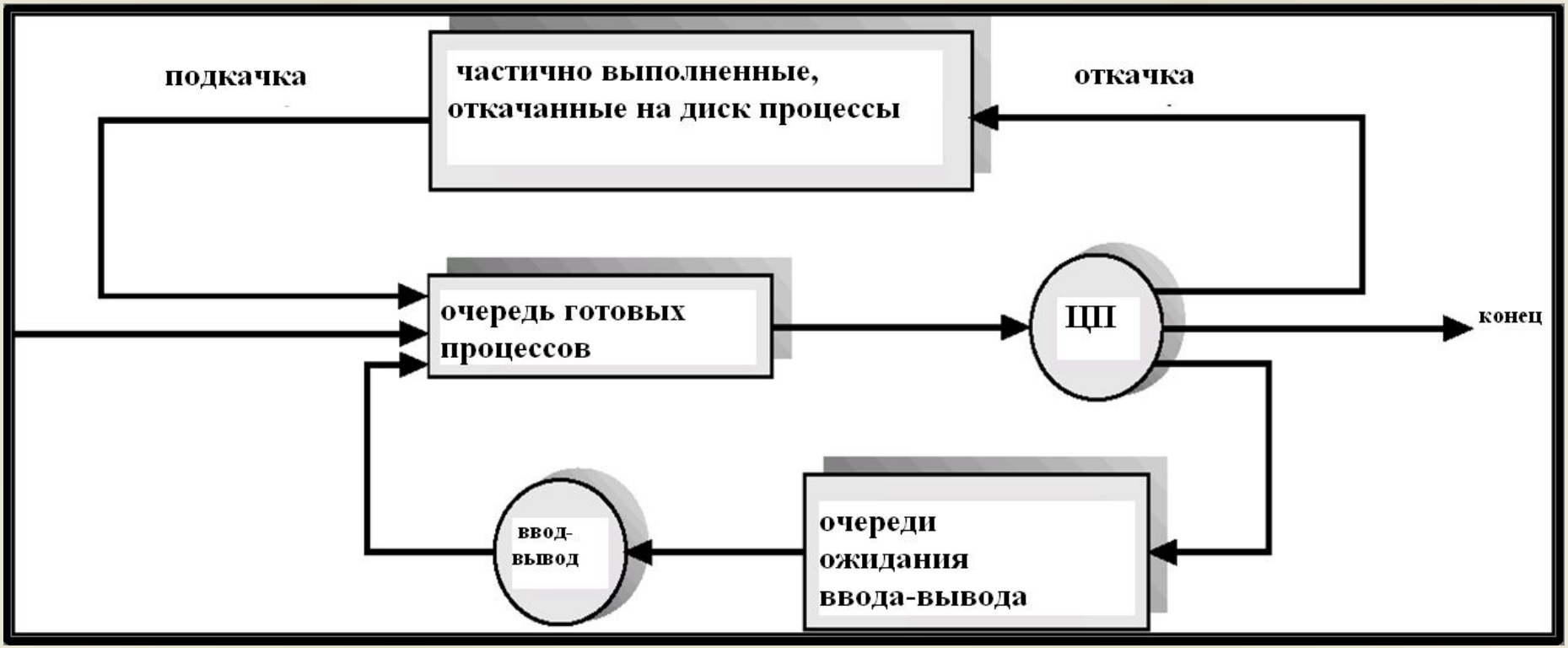
Графическое представление диспетчеризации процессов



Планировщики

- *Долговременный планировщик* (планировщик заданий) – определяет, какие процессы должны быть перемещены в очередь готовых процессов
- *Кратковременный планировщик* (планировщик процессора) – определяет, какие процессы должны быть выполнены следующими и каким процессам должны быть предоставлены процессоры.

Добавление планировщика откачки и подкачки процессов



Особенности планировщиков и процессов

- Кратковременный планировщик вызывается очень часто (в течение ближайших миллисекунд) => должен быть очень быстрым
- Долговременный планировщик вызывается относительно редко (минуты, секунды) => может быть сравнительно медленным
- Именно долговременный планировщик определяет *степень (коэффициент) мультипрограммирования*
- Процессы можно описать как:
  - Ориентированные на ввод-вывод (I/O-*bound*) – тратят больше времени на ввод-вывод, чем на вычисления; расходуют много коротких квантов процессорного времени
  - *Ориентированные на использование процессора (CPU-bound*) – тратят основное время на вычисления; расходуют небольшое число долговременных квантов процессорного времени

Переключение контекста процесса (context switch)

- Когда процессор переключается на другой процесс, система должна сохранить состояние старого процесса и загрузить сохраненное состояние для нового процесса
- Переключение контекста относится к накладным расходам (overhead); система не выполняет никаких полезных действий при переключении с одного процесса на другой
- Время зависит от аппаратной поддержки.
- Пример: “Эльбрус” – контекстное переключение – одна команда *СМСТЕК* (сменить стек, т.е. переключиться с одного облегченного процесса на другой)

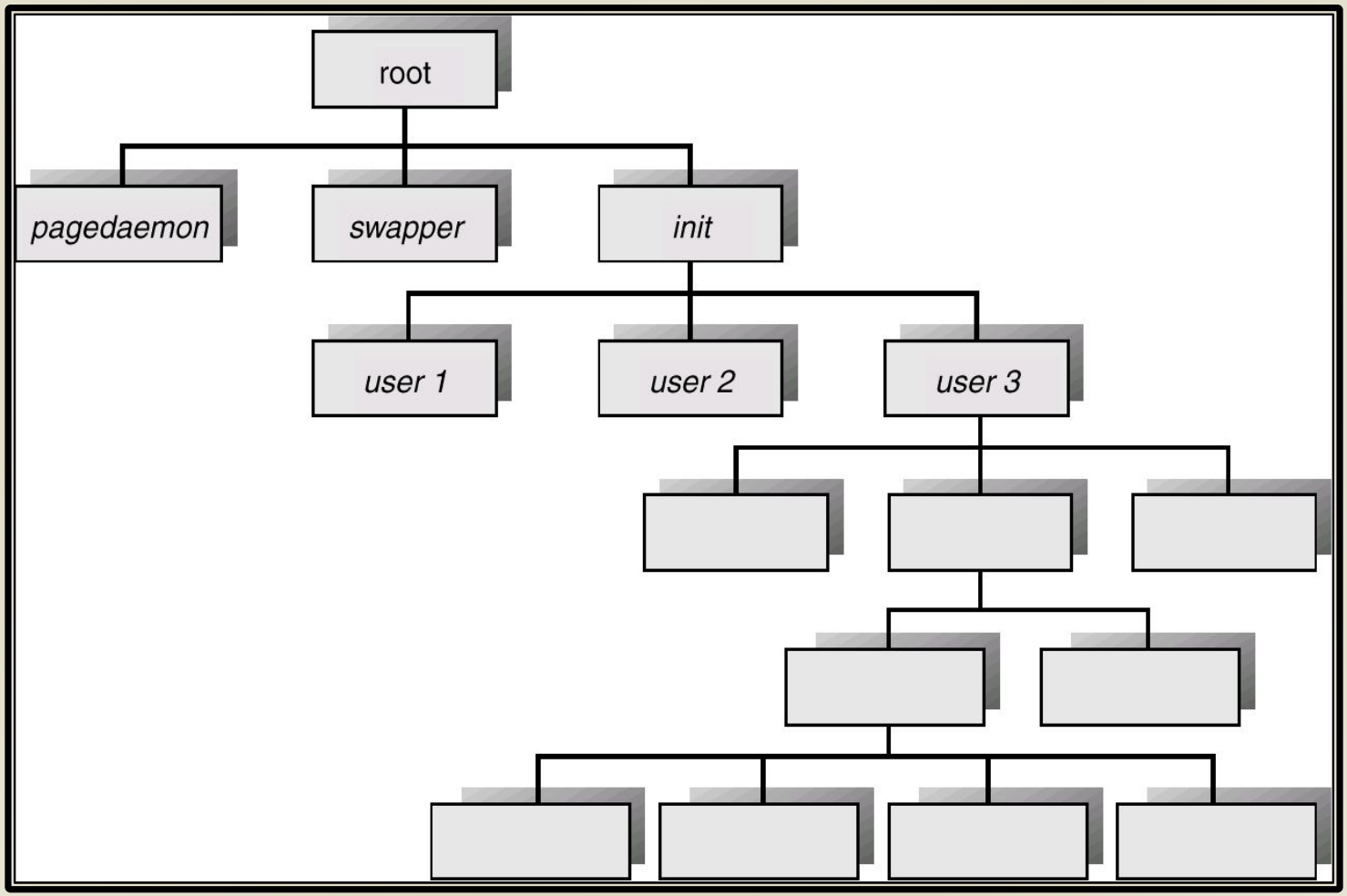
Создание процесса

- Процесс-родитель создает дочерние процессы, которые, в свою очередь, создают другие процессы, тем самым формируя *дерево процессов*
- Разделение ресурсов
  - Процесс-родитель и дочерние процессы разделяют все ресурсы
  - Дочерние процессы разделяют подмножество ресурсов процесса-родителя
  - Процесс-родитель и дочерний процесс не имеют общих ресурсов
- Исполнение
  - Процесс-родитель и дочерние процессы исполняются совместно
  - Процесс-родитель ожидает завершения дочерних процессов

Адресация и создание процесса

- Адресное пространство
  - Дочернего процесса копирует адресное пространство процесса-родителя
  - У дочернего процесса имеется программа, загруженная в него
- UNIX:
  - fork – системный вызов, создающий новый процесс
  - exec (execve) – системный вызов, используемый после fork, с целью замены пространства памяти процесса новой программой

Дерево процессов в системе UNIX



Уничтожение процесса

- Процесс исполняет заключительный оператор и обращается к ОС для своей ликвидации (exit).
  - Передача данных от дочернего процесса процессу-родителю (wait).
  - Ресурсы процесса освобождаются операционной системой
- Процесс-родитель может уничтожить дочерние процессы (abort).
  - Дочерний процесс превысил выделенные ему ресурсы
  - Решения задачи, порученной дочернему процессу, больше не требуется
  - Происходит выход из процесса-родителя
    - ОС не допускает продолжения исполнения дочернего процесса, если его процесс-родитель уничтожается
    - “Каскадное” уничтожение процессов