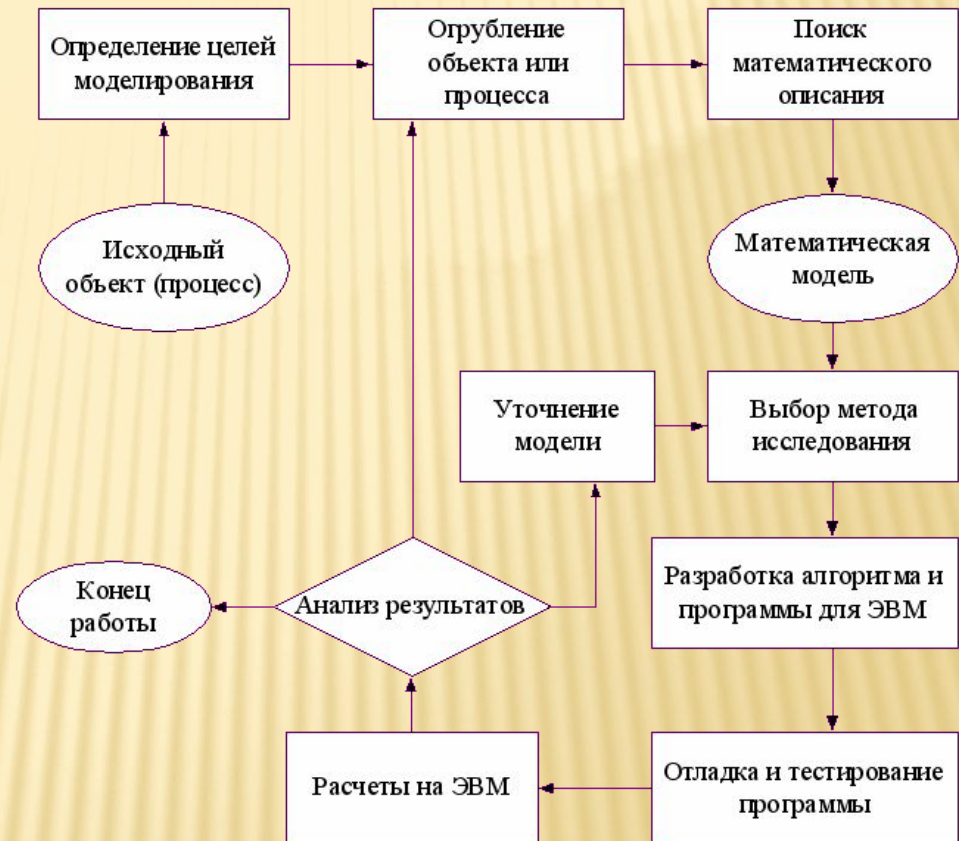


ОБЗОР ПРОЦЕССА СОЗДАНИЯ ДОЧЕРНИХ ПРОЦЕССОВ

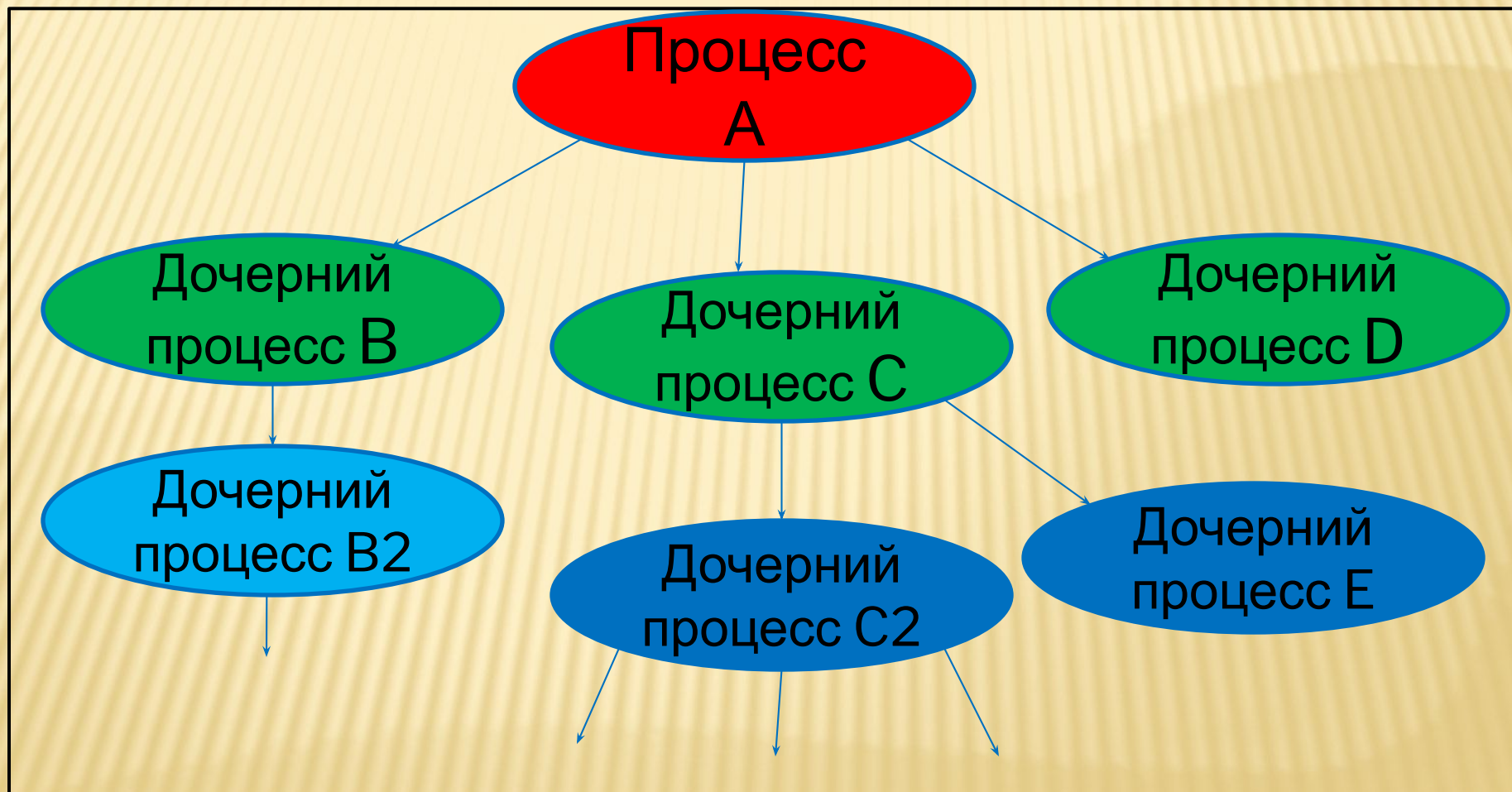
Павлов Г.Б.
2ивт-Здб-280

ПРОЦЕСС

- программа, которая выполняется в текущий момент. Стандарт ISO 9000:2000 определяет процесс как совокупность взаимосвязанных и взаимодействующих действий, преобразующих входящие данные в исходящие.



ИЕРАРХИЯ ПРОЦЕССА



ИСПОЛЬЗОВАНИЕ РЕСУРСОВ ПРИ СОЗДАНИИ ПРОЦЕССА

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.

Адресация и использование памяти.

- Адресное пространство дочернего процесса копирует адресное пространство процесса-родителя; у дочернего процесса имеется программа, загруженная в него;
- Дочерний процесс выполняется в том же пространстве памяти, что и процесс-родитель (облегченный процесс).

Исполнение.

- Процесс-родитель и дочерние процессы выполняются совместно
- Процесс-родитель ожидает завершения дочерних процессов

СОЗДАНИЕ ДОЧЕРНЕГО ПРОЦЕССА

- `fork` - создающий новый процесс (потомок), который является практически полной копией процесса-родителя, выполняющего этот вызов.
- `vfork` функция подобна `fork`, но более эффективна.
- В Windows новый процесс создается функцией `CreateProcess`

СЕМЕЙСТВО ФУНКЦИЙ EXEC

`execv`

`execve`

`execvp`

`exec1`

`execle`

`exec1p`

Эти функции используются, чтобы заставить дочерний процесс выполнить новую программу после того, как он был раздвоен.

ПРОЦЕСС-СИРОТА

в семействе операционных систем UNIX вспомогательный процесс, чей основной процесс (или связь с ним) был завершён нештатно (не подав сигнала на завершение работы).



ПРОЦЕСС-ЗОМБИ



дочерний процесс в Unix-системе, завершивший своё выполнение, но ещё присутствующий в списке процессов операционной системы, чтобы дать *родительскому* процессу считать код завершения.

ЗАВЕРШЕНИЕ ПРОЦЕССА

- входная функция первичного потока возвращает управление
- один из потоков процесса вызывает функцию *ExitProcess*
- поток другого процесса вызывает функцию *TerminateProcess*
- все потоки процесса умирают по своей воле

ПРИ ЗАВЕРШЕНИИ ПРОЦЕССА:

- Выполнение всех потоков в процессе прекращается
- Все объекты, созданные процессом, уничтожаются, а объекты ядра закрываются (если их не использует другой процесс).
- Код завершения процесса меняется со значения `STILL_ACTIVE` на код, переданный в `ExitProcess` или `TerminateProcess`.
- Объект ядра "процесс" переходит в свободное, или незанятое (`signaled`), состояние. Прочие потоки в системе могут приостановить свое выполнение вплоть до завершения данного процесса.
- Счетчик объекта ядра "процесс" уменьшается на 1

**Спасибо за
внимание**