ЛР2 по ОСиС, Андреев Александр ИУ7-54Б 09.11.2021, Вторник

(I.)fork()

1. Для данного и стека процесса-потомка резервируется пространство;
2. Назначаются:
   - идентификатор процесса PID
   - структура ркос потомка;
3. Инициализируется структура ркос потомка (Некоторые поля копируются от процесса родителя: ~~инде~~ идентификаторы пользователей и групп, маска сигналов и группа процессов. Часть полей инициализ-ся ф. Часть полей инициализ-ся след-ми для потомка значениями:
   - PID потомка и его родителя
   - указатель на структуру ркос родителя;
4. Создаются карты ~~стр~~ трансляции адресов для процесса-потомка;
5. Вдается область 'и' потомка и в нее копи-ся область 'и' процесса предка;
6. Изменяются ссылки области 'и' на новые карты адресации и прос-во свопинга;
7. Потомок добавляется в набор процессов, которые разделяют область кода процесса, вып-ой процессом-родителем;
8. Копируются дублируются области данных и стека родителя, ~~которые он наследует~~ ~~открытые файлы~~ (потомок ~~наследует~~ дескрипторы) ~~текущий рабочий каталог~~;
9. Потомок получает ссылки на разделяемые ресурсы, которые он наследует: открытые файлы (потомок наследует дескрипторы) и текущий рабочий каталог;
10. Инициализ-ся аппаратный контекст потомка путем копирования регистров родителя;
11. Помещать процесс-потомок в очередь готовых процессов;
12. Возвращается PID в точку возврата из системного вызова в родительском процессе и 0 – в процессе-потомке.

(II.) exec()

1. Разбирает путь к исп-му файлу и осущ-ет доступ к нему;
2. Проверяет, имеет ли вызв-ый процесс полномочия на вып-е файла;
3. Читает заголовок, проверяет, что он исполняемый;
4. Если для файла уст-ны биты SUID/SGID, то эффективные идент-ры UID и GID вызв-го процесса изменяет на UID и GID, соо-ие владельцу файла;
5. Копирует аргументы, перед-е в exec, переменные среды в пространство ядра, после чего текущее польз-е пр-во готово к уничтожению;
6. Выделяет пр-во свопинга для области данных и стека;
7. Высв-ет старое адресное прост-во и связанное с ним прост-во свопинга. Если же процесс был создан при помощи vfork, производится возврат старого адресного прост-ва роди-му процессу.
8. Выделяет карты трансляции адресов для новых текста, данных и стека;
9. Устанавливает новое адресное прост-во. Если область текста активна (какой-то другой адрес уже вып-ет програ-му), то она будет совместно исп-ся с этим процессом. В других случаях прост-во должно инициал-ся из вспомогательного файлового процесса в них разбита на страницы: каждая страница считыв-ся в память по мере необходимости;
10. Копирует аргументы и переменные ~~ядра~~ среды обратно в новый сеанс приложения.
11. Сбрасывает все обраб-ки сигналов в действие, опр-е пользо-ю, так как функции обработчиков сигналов не существуют в новой программе. Сигналы, которые были проигнорированы или забл-ны перед вызовом exec, сбр-ся в те же состояния.
12. Инициализируется аппаратный контекст. При этом большинство регистров сбрасываются на 0, а указателю команд присваивается значение точки входа программы.