1. Процессы и потоки

Процесс - экземпляр замущенного исполняемого файла. Нужен для того, что абстрагивровать писание программы, как будто все возможности только твои.

Процесс состоит из **потоков**. Это вычислительный контекст. Раньше он обычно был один, потому что у нас было только одно ядро. Но со временем появились ядра и появилась необходимость писать программу в несколько потоков.

1.1. Что общего у потоков?

1) регистры

2) стек

3) ид потока

4) IPC

1.2. Как выглядит память у потока?

stack - сохранение регистров
|
V
.....
|
heap - в c++(free store) - динамическая память

_____
data - статик переменные

_____
text - код программы

1.3. Действия с процессами

1.3.1. fork

Это создание нового процесса. До форка был один процесс(1). После стало два процесса и все стало параллельно

Пример кода для использования:

Но понятное дело, что писать так постоянно грустно и произойдет *fork bomba*

1.3.2. execve

Заместить наш процесс новым процесс по имени с аргументами

Пример - мы хотим вызвать man из bash-a

1.3.3. waitpid

Подождать конец процесса с ид . Например в предыдущем коде можно внутри написать waitpid - подождать пока выйдет ребенок

1.3.4. Exit and kill

Убить себя или чужой процесс с состоянием . Если хочешь завершить чужой процесс - передаешь с которым ты хочешь завершить процесс

1.4. Виды процессов

1.4.1. Процесс 1

Ядро всегда создает процесс с номером 1. У нее есть две специальные команды для системы инициализации. Примером может быть **init/systemd**. Они запускают системные сервисы.

1.4.2. Zombie процессы

Что будет если не сделать **waitpid** у родителя? После завершения процесса он освобождается, но если не вызвать он будет "зомби-процессом будет занимать немного места, но он будет щанимать ид-шники. Также его нельзя убить.

1.5. Orphan(сирота)

Ситуация когда родителя убили. Пид родителя становится 1.

1.6. Scheduler

Как процессы живут вместе? Давайте постепенно переключать процессы. Этим занимается ОС с помощью scheduler. Он берет и с помощью некоторого порядка задает очередь выполнения процессов. Каждый процесс работает по определенному косучку времени(например 15 мс).