Sys call

* fork()
* exit()
* wait()
* kill(pid)
* getpid()
* sleep(n)
  + n초간 틱을 전부 sleep하는 것?
  + n번의 틱을 전부 sleep하는 것?(이 경우 의미가 없어보임)
* xec(filename, \*argv)
* sbrk(n)
* open(filename, flags)
* \_read(fd, buf, n)
* write(fd, buf, n)
* close(fd)
* dup(fd)
* pipe(p)
* chdir(dirname)
* mkdir(dirname)
* mknod(name, major, minor)
* fstat(fd)
* link(fillename)
* unllink(filename)

exec

* xv6는 ELF(Executable and Linkable Format) 형식으로 이루어져 있다.

개인 질문

Sys call ) dup이 뭐하는 거지 ?

Pipe는 어디에 존재하는 애지?

runcmd가 뭐지?

unlink가 파일 delete하는 거랑 뭐가 달라?

선배님께 질문

sbrk

* 알기로는 malloc은 sbrk를 호출하지 않고 이미 할당받은 메모리 중에서 일부를 넘겨 준다고 들었습니다. 그러면, 시스템이 sbrk를 해놓는 단위에 대해서 알고 싶습니다.
  + 만약 처음부터 너무 많이 해놓으면 heap영역을 많이 사용하지 않는 프로그램의 경우 메모리 손해라고 생각되고
  + 아닐 경우 너무 빈번한 sbrk의 호출이 일어날 것으로 예상
  + 아님아님
    - Heap에 할당된 메모리들은 이미 할당된 애들일 테니까 불려올 때, sbrk라는 syscall을 쓸필요가 없을 것 같은데
    - 그러면 sbrk는 맨처음에 프로세스를 만들 때 사용되나?
      * 이건 exec에서 만들어 지는 것 아닌가?
        + 그럼 exec만으로 부족한 프로세스의 경우에 sbrk로 만듬?