Assignment 3 Adv
Advance Assignment

- Shared pages and copy-on-write
- sbrk()
- Demand loading and mmap
- Paging
Shared pages and Copy-on-write

• What are they
• Why are they useful
• What they are not
  – Shared memory
Two (or more) processes running the same program and sharing a section.
COW

- fork() can be more efficient
- as_copy is underlying routine
- set pages read_only
  - Keep reference count in frame table
  - On write-fault, vm_fault copies, decrement count.
The "break" is the end address of a process's heap region.

The sbrk call adjusts the "break" by the amount.

It returns the old "break". Thus, to determine the current "break", call sbrk(0).

The heap region is initially empty, so at process startup, the beginning of the heap region is the same as the end and may thus be retrieved using sbrk(0).
mmap() and demand loading
Memory-mapped files and paging
mmap/munmap semantics

void *mmap(size_t length, int prot, int fd, off_t offset);
int munmap(void *addr);
void *mmap(size_t length, int prot, int fd, off_t offset);
munmap

int munmap(void *addr);
demand loading

Executable file

- code
- data
- heap
- stack
- Free RAM
- OS/161 Kernel
- code
- data
- code