

## Assignment 3 Adv

1

## Advance Assignment

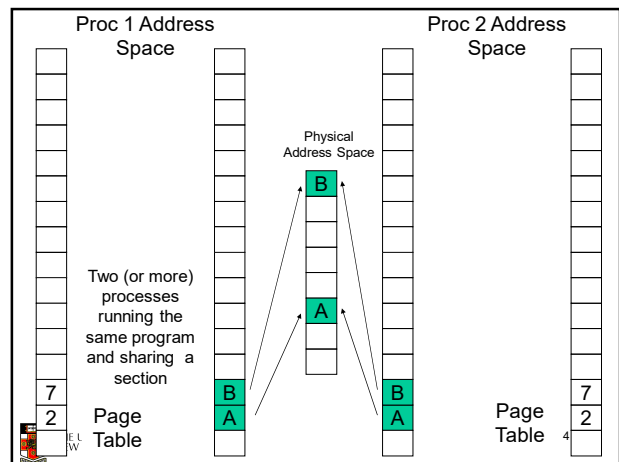
- Shared pages and copy-on-write
- sbrk()
- Demand loading and mmap
- Paging

2

## Shared pages and Copy-on-write

- What are they
- Why are they useful
- What they are not
  - Shared memory

3



4

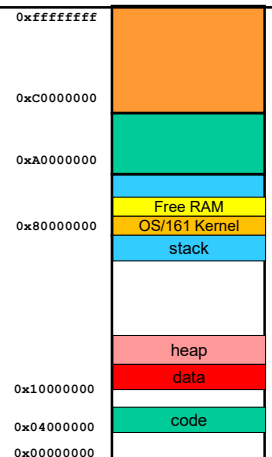
## COW

- 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.

5

## sbrk

- 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).

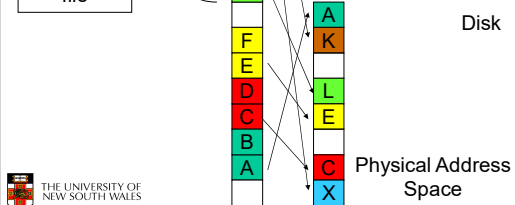


6

# mmap() and demand loading

# Memory-mapped files and paging

Memory mapped file

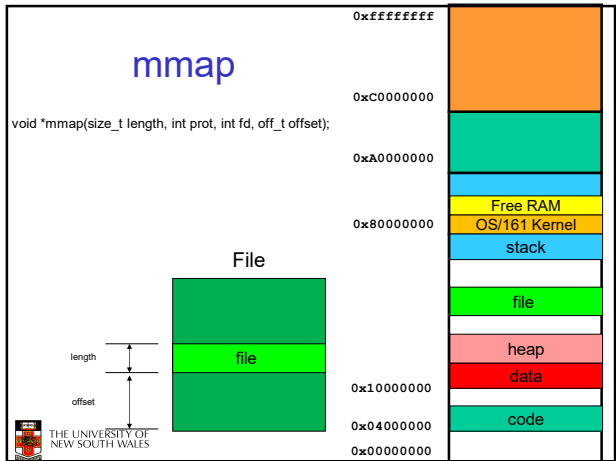


# mmap/munmap semantics

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

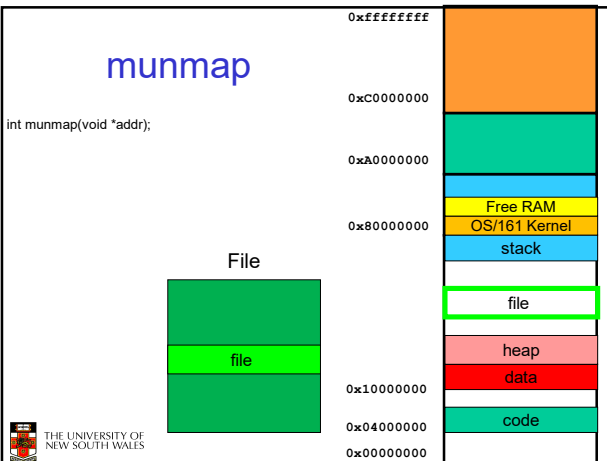
# mmap

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



# munmap

```
int munmap(void *addr);
```



# demand loading

