Virtual Memory Lab
Archive Printer
Quiz

- Address Spaces
- Segmentation and Paging
Copy-On-Write (COW)

- fork() creates a duplicate copy of parent address space.
- Child invokes exec() that overwrites the copied address space.
  - Copying address space c is unnecessary.
- COW
  - Parent and child processes share the same pages of the address space.
  - If parent or child writes into a shared page (marked as COW), a copy of the shared page is created.
Copy-On-Write (COW)

Before process 1 modifies page C.

After process 1 modifies page C.
mmap

void* mmap (p, CHUNK_SIZE, PROT_READ | PROT_WRITE, MAP_ANONYMOUS | MAP_SHARED, -1, 0)

p: The address of a new mapping in the address space
CHUNK_SIZE: The length of mapping
PROT_READ | PROT_WRITE: make mapping readable and writable
MAP_ANONYMOUS: Mapping is not associated with some file
MAP_SHARED: Updating the contents of mapping is visible to other processes
mremap

void* mremap (chunk, 0, CHUNK_SIZE, MREMAP_MAYMOVE)

chunk: The address of an existing memory space to create a duplicate mapping from

CHUNK_SIZE: The length of the duplicate mapping

MREMAP_MAYMOVE: The duplicate mapping can be relocated to another location in the address space
mprotect

Int mprotect (region_start, region_size, PROT_READ)

region_start: The start of the region for setting protection
region_size : The length of the region that is protected
PROT_READ: The type of protection