

System calls:

Int get_share_mem(int id, char* va) :

Maps the shared memory corresponding to id, to virtual address starting from va. va must be page-aligned.

If no shared memory with given id exists, the call fails.

Returns 1 on success, 0 on failure.

Int create_share_mem(int id, char* va)

Creates a new shared memory region of size 4096 bytes, and maps it virtual address va. va must be page-aligned. If id is -1, then the first unallocated id is assigned. If id is non-zero, creates a memory region with given id, if a region already exists with given id then it returns -1.

API's:

char* get_pa_sharemem(int id):

Returns the physical address of the shared memory corresponding to given id. Panics if no memory with corresponding id exists.

Bool is_availabe_sharemem(int id):

Returns whether the given id is being used by any other process for its shared memory. Returns True if no other process is using the id.

Int get_id_sharemem(char * va):

Returns the id corresponding to the physical address pointed at by the given virtual memory. Panics if the physical address isn't mapped to any shared memory.

Void free_sharemem(char* v, pde_t pgdir):

If no other process is using the this shared memory, deallocate it. Panics if the va doesn't correspond to any shared memory.

Test Case:

```
void
share_mem(void)
{
    char *va = (char*) 0x40960000;
    int id = 4;
    if( get_share_mem(id, va) > 0){
        goto failed;
    }
    if( create_share_mem(id, va) < 0){
        goto failed;
    }

    for(int i =0;i<4095; i++){
```

```

    *va = '1';
    va++;
}

if(fork()==0){
    char *va2 = (char*) 0x20480000;

    if(get_share_mem(id,va2) < 0){
        goto failed;
    }
    for(int i = 0;i<2048;i++){
        if(*va2 != '1'){
            goto failed;
        }
        va2++;
    }

    for(int i = 2048; i < 4095; i++){
        *va2 = '2';
        va2++;
    }
    printf(1, "child done");
    exit();
}

else{
    wait();
    va = (char*) 0x40960000;
    for(int i = 0;i<2048;i++){
        if(*va != '1'){
            goto failed;
        }
        va++;
    }

    for(int i = 2048; i < 4095; i++){
        if(*va != '2'){
            goto failed;
        }
        va++;
    }
}

printf(1, "share_mem ok %d\n", bstat());
exit();
failed:

```

```
    printf(1, "test failed!\n");  
    exit();  
}
```