Here’s the complete content for a **worksheet on System V Shared Memory and Record Locking**, organized in a copy-paste-friendly format. You can paste this into **Word** or **any text editor** and save/export as DOCX or PDF.

**📘 System V Shared Memory – Worksheet**

**Section A: Objective Type (Choose the correct answer)**

1. Which header file is required for System V shared memory?  
   a) <sys/ipc.h>  
   b) <sys/shm.h>  
   c) <sys/types.h>  
   d) All of the above
2. Which function is used to create or get a shared memory segment?  
   a) shmget()  
   b) shmctl()  
   c) shmat()  
   d) shmdt()
3. What is the return type of shmat()?  
   a) int  
   b) void \*  
   c) char \*  
   d) long
4. What does IPC\_RMID do in shmctl()?  
   a) Removes the segment immediately  
   b) Marks the segment to be destroyed after detachment  
   c) Attaches the segment to process space  
   d) None of the above
5. Which function detaches a shared memory segment from process space?  
   a) shmctl()  
   b) shmat()  
   c) shmdt()  
   d) shmget()

**Section B: Fill in the Blanks**

1. The function used to attach a shared memory segment is \_\_\_\_\_\_\_\_\_\_\_\_.
2. The function used to mark a shared memory segment for removal is \_\_\_\_\_\_\_\_\_\_\_\_.
3. The prototype of shmget() is \_\_\_\_\_\_\_\_\_\_\_\_.
4. The key used in shmget() is generated using \_\_\_\_\_\_\_\_\_\_\_\_.
5. A shared memory segment can be shared between \_\_\_\_\_\_\_\_\_\_.

**Section C: Programming Exercise**

**Q11.** Write a program to:

* Create a shared memory segment
* Attach it
* Write the text **"Shared memory test"** into it

**📕 System V Record Locking – Worksheet**

**Section A: Objective Type (Choose the correct answer)**

1. Which system call is used to apply a lock on a file record?  
   a) open()  
   b) fcntl()  
   c) lockf()  
   d) read()
2. What is the name of the structure used for file locking?  
   a) struct flock  
   b) struct lock  
   c) struct filelock  
   d) struct file
3. What does F\_SETLK do in fcntl()?  
   a) Gets the lock status  
   b) Sets a lock (non-blocking)  
   c) Waits to get a lock  
   d) Clears the lock
4. Which member of struct flock indicates the type of lock?  
   a) l\_whence  
   b) l\_len  
   c) l\_type  
   d) l\_pid
5. What does F\_UNLCK represent?  
   a) Unlock a locked region  
   b) Update file permission  
   c) Set read lock  
   d) None of the above

**Section B: Fill in the Blanks**

1. \_\_\_\_\_\_\_\_\_\_\_\_ is used to apply a blocking lock on a file.
2. \_\_\_\_\_\_\_\_\_\_\_\_ is used to unlock a region in a file.
3. l\_type can be \_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_, or \_\_\_\_\_\_\_\_\_\_\_\_.
4. l\_start in struct flock indicates \_\_\_\_\_\_\_\_\_\_\_\_.
5. File locks are maintained at \_\_\_\_\_\_\_\_\_\_\_\_ level.

**Section C: Programming Exercise**

**Q11.** Write a program to:

* Open a file
* Apply a **write lock** using fcntl()
* Display lock status

Here is the **Answer Key** for the **System V Shared Memory and Record Locking Worksheet**:

**✅ System V Shared Memory – Answer Key**

**Section A: Objective Type**

1. **d)** All of the above
2. **a)** shmget()
3. **b)** void \*
4. **b)** Marks the segment to be destroyed after detachment
5. **c)** shmdt()

**Section B: Fill in the Blanks**

1. **shmat()**
2. **shmctl() with IPC\_RMID**
3. **int shmget(key\_t key, size\_t size, int shmflg);**
4. **ftok()**
5. **Multiple processes**

**Section C: Programming Exercise – Sample Code**

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <string.h>

int main() {

key\_t key = ftok("shmfile", 65);

int shmid = shmget(key, 1024, 0666|IPC\_CREAT);

char \*str = (char\*) shmat(shmid, (void\*)0, 0);

strcpy(str, "Shared memory test");

printf("Data written in memory: %s\n", str);

shmdt(str);

return 0;

}

**✅ System V Record Locking – Answer Key**

**Section A: Objective Type**

1. **b)** fcntl()
2. **a)** struct flock
3. **b)** Sets a lock (non-blocking)
4. **c)** l\_type
5. **a)** Unlock a locked region

**Section B: Fill in the Blanks**

1. **F\_SETLKW**
2. **F\_UNLCK**
3. **F\_RDLCK, F\_WRLCK, F\_UNLCK**
4. **The byte offset in the file to start the lock**
5. **Process**

**Section C: Programming Exercise – Sample Code**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

int main() {

int fd = open("test.txt", O\_WRONLY | O\_CREAT, 0666);

struct flock lock;

lock.l\_type = F\_WRLCK;

lock.l\_whence = SEEK\_SET;

lock.l\_start = 0;

lock.l\_len = 0;

lock.l\_pid = getpid();

if (fcntl(fd, F\_SETLK, &lock) == -1) {

perror("Lock failed");

} else {

printf("Write lock applied\n");

}

getchar(); // Wait to release lock

lock.l\_type = F\_UNLCK;

fcntl(fd, F\_SETLK, &lock);

printf("Lock released\n");

close(fd);

return 0;

}