

# LABSHEET 7.1 Shared Memory

## 1) . Modify the child to sleep(2)

- Result: The parent writes more frequently than the child reads. So, when the child reads, it might skip some intermediate values written by the parent.
- Output Behavior: Child's prints may look like sudden jumps in values because some writes happened while it was sleeping.

## 2) Modify the parent to sleep(2) and child back to sleep(1)

- Result: The child reads more frequently than the parent writes.
- Output Behavior: Child often sees repeated values between updates.

## 3)

```
#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <unistd.h>

#include <sys/wait.h>

int main() {

int shmid;

int *n;

shmid = shmget(IPC_PRIVATE, sizeof(int), 0666 | IPC_CREAT);

if (shmid == -1) {

perror("shmget failed");

return 1;

}

if (fork() == 0) {

sleep(1);

= (int *)shmat(shmid, NULL, 0);

printf("Child: Odd numbers till %d are:\n", *n);

for (int i = 1; i <= *n; i += 2) {
```

```

printf("%d ", i);
}
printf("\n");
shmdt(n);
} else {
// Parent process
n = (int *)shmat(shmid, NULL, 0);
printf("Parent: Enter an integer: ");
scanf("%d", n); // input happens here
wait(NULL); // wait for child to complete
shmdt(n);
shmctl(shmid, IPC_RMID, NULL);
}
return 0;
}

```

```

Parent: Enter an integer: Child: Odd numbers till 0 are:
10

```

4)

```

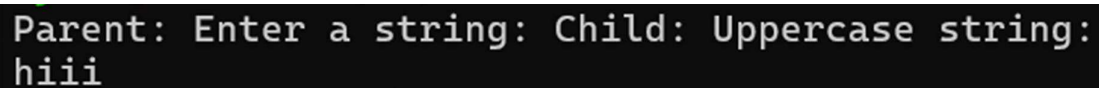
#include <stdio.h>
#include <ctype.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>
#include <sys/wait.h>
#include <string.h>
#define SIZE 100
int main() {
int shmid;
char *shm;
shmid = shmget(IPC_PRIVATE, SIZE, 0666 | IPC_CREAT);
if (shmid == -1) {

```

```

perror("shmget failed");
return 1;
}
if (fork() == 0) {
sleep(1); // Ensure parent writes first
shm = (char *)shmat(shmid, NULL, 0);
for (int i = 0; shm[i] != '\0'; i++) {
shm[i] = toupper(shm[i]);
}
printf("Child: Uppercase string: %s\n", shm);
shmdt(shm);
} else {
// Parent
shm = (char *)shmat(shmid, NULL, 0);
printf("Parent: Enter a string: ");
fgets(shm, SIZE, stdin);
shm[strcspn(shm, "\n")] = '\0'; // Remove newline
wait(NULL); shmdt(shm);
shmctl(shmid, IPC_RMID, NULL);
}
return 0;
}

```



```

Parent: Enter a string: Child: Uppercase string:
hiii

```

5)

```

#include <stdio.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>
#include <stdlib.h>

```

```

#define SIZE 100

int main(int argc, char *argv[]) {
    key_t key = 1234;

    int shmid;

    char *shm;

    shmid = shmget(key, SIZE, 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget");
        return 1;
    }

    shm = (char *)shmat(shmid, NULL, 0); if (argc != 2) {
        printf("Usage: %s [1|2|3]\n", argv[0]);
        return 1;
    }

    int process_id = atoi(argv[1]);
    if (process_id == 1) {
        while (shm[0] != '3') sleep(1);
        printf("Process 1 reads: %s\n", shm + 1);
    } else if (process_id == 2) {
        shm[0] = '2';
        strcpy(shm + 1, "hello ");
    } else if (process_id == 3) {
        while (shm[0] != '2') sleep(1);
        strcat(shm + 1, "memory");
        shm[0] = '3';
    }

    shmdt(shm);

    if (process_id == 1) shmctl(shmid, IPC_RMID, NULL);
    return 0;
}

```

```
Usage: ./a.out [1|2|3]
```