Tab 1

**OS LAB04**

**Fork Command**

---------------------------------------------------------------------------------------------------------------------------

**LAB Work:**  
Fork system call

Fork return PID

getpid(), getppid()

Zombie process vs Orphan process

---------------------------------------------------------------------------------------------------------------------

**Fork Working**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

int main(){

printf("Hello\n"); // always use \n with printf to clear the buffer

fork();

fork();

fork();

printf("Hi\n");

return 0;

}

**Fork with return value**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

int main(){

printf("Hello\n"); // always use \n with printf to clear the buffer

int pid = fork();

if(pid == -1){

return 1;

}

else if(pid == 0)

printf("Hi\n");

return 0;

}

**\*Fork with Wait**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(){

int pid = fork();

if(pid == -1)

return -1;

else if(pid != 0){

wait(NULL);

printf("Hello from Parent: %d having parent: %d\n", getpid(), getppid());

}else

printf("Hello from Child: %d having parent: %d\n", getpid(), getppid());

return 0;

}

---------------------------------------------------------------------------------------------------------------------

**\*Fan of n-processes**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(){

int n;

printf("Give input of number of childs: ");

scanf("%d", &n);

printf("Parent processes PID: %d\n\n", getpid());

for(int i=0; i<n; i++){

int pid = fork();

if(pid == -1)

return 1;

else if(pid == 0){

printf("Child process: %d with parent: %d\n", getpid(), getppid());

break;

}

else

wait(NULL);

}

printf("Bye from process: %d\n\n", getpid());

return 0;

}

---------------------------------------------------------------------------------------------------------------------

**\*N-processes Linked List**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(){

int n;

printf("Give input of number of nodes in list: ");

scanf("%d", &n);

printf("1st process PID: %d\n\n", getpid());

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

int pid = fork();

if(pid == -1)

return 1;

else if(pid == 0){

printf("Child process: %d with parent: %d\n", getpid(), getppid());

}

else{

wait(NULL);

break;

}

}

printf("Bye from process: %d\n", getpid());

return 0;

}

---------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------