**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**

****

**A Lab Report #3**

**[ Course title: COMP 307]**

**Submitted by:**

**Bisheshwor Neupane(35)**

**Submitted to:**

**Mr. Dhiraj Shrestha**

**Department of Computer Science and Engineering**

**Submission Date:**

**August 06, 2021**

**(Understanding Process Concepts)**

1. Write a C program to implement process System calls ? [Hint use fork())]
2. How many processes are created in a given program?

#include <stdio.h>

#include <unistd.h>

int main()

{

int i;

for (i = 0; i < 4; i++)

fork();

return 0;

}

1. When will line J be reached?

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main()

{

pid\_t pid;

/\* fork a child process \*/

pid = fork();

if (pid < 0) { /\* error occurred \*/

fprintf(stderr, "Fork Failed");

return 1;

}

else if (pid == 0) { /\* child process \*/

execlp("/bin/ls","ls",NULL);

printf("LINE J");

}

else { /\* parent process \*/

/\* parent will wait for the child to complete \*/

wait(NULL);

printf("Child Complete");

}

return 0;

}

1. What are the pid values?

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main()

{

pid\_t pid, pid1;

/\* fork a child process \*/

pid = fork();

if (pid < 0) { /\* error occurred \*/

fprintf(stderr, "Fork Failed");

return 1;

}

else if (pid == 0) { /\* child process \*/

pid1 = getpid();

printf("child: pid = %d",pid); /\* A \*/

printf("child: pid1 = %d",pid1); /\* B \*/

}

else { /\* parent process \*/

pid1 = getpid();

printf("parent: pid = %d",pid); /\* C \*/

printf("parent: pid1 = %d",pid1); /\* D \*/

wait(NULL);

}

return 0;

}

1. What will be at Line X and Line Y?

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

#define SIZE 5

int nums[SIZE] = {0,1,2,3,4};

int main()

{

int i;

pid t pid;

pid = fork();

if (pid == 0) {

for (i = 0; i < SIZE; i++) {

nums[i] \*= -i;

printf("CHILD: %d ",nums[i]); /\* LINE X \*/

}

}

else if (pid > 0) {

wait(NULL);

for (i = 0; i < SIZE; i++)

printf("PARENT: %d ",nums[i]); /\* LINE Y \*/

}

return 0;

}