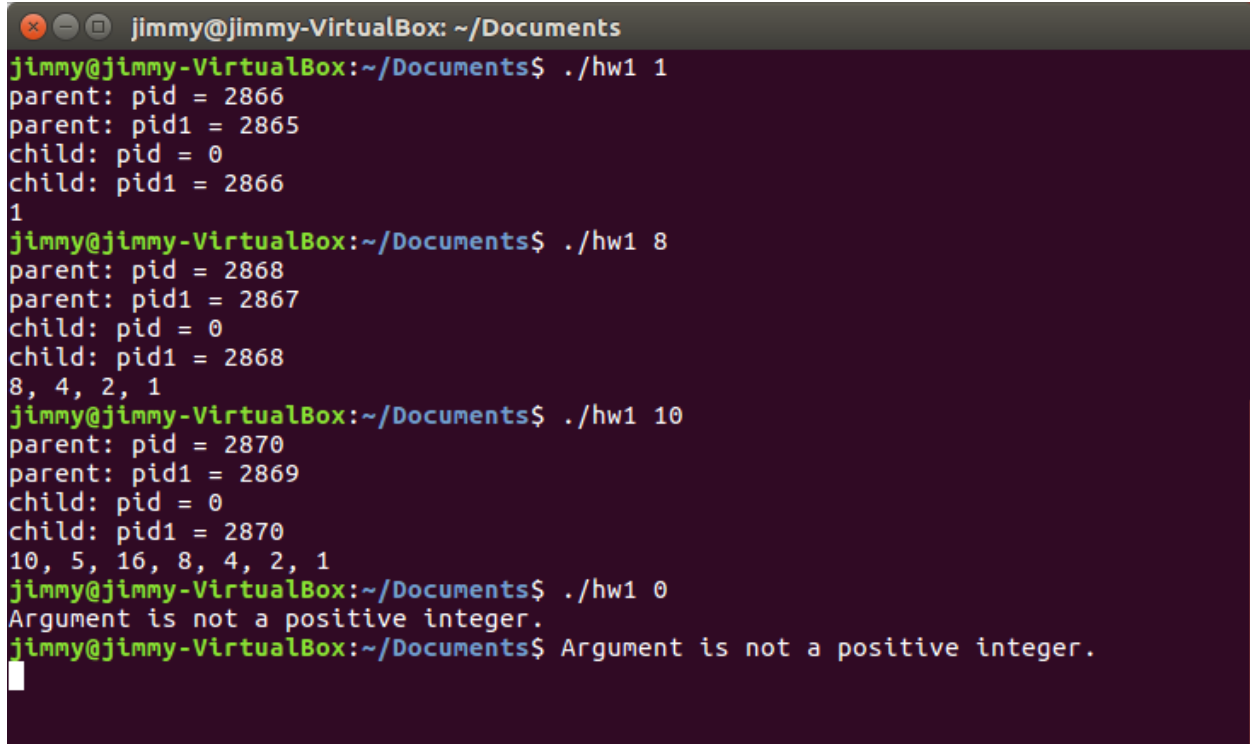


Homework 1 Report
CSC 4320 Operating Systems
Spring 2018
Name: Jimmy Tran
Email: jtran25@student.gsu.edu

1. Screenshot of program outputs.



```
jimmy@jimmy-VirtualBox: ~/Documents
jimmy@jimmy-VirtualBox:~/Documents$ ./hw1 1
parent: pid = 2866
parent: pid1 = 2865
child: pid = 0
child: pid1 = 2866
1
jimmy@jimmy-VirtualBox:~/Documents$ ./hw1 8
parent: pid = 2868
parent: pid1 = 2867
child: pid = 0
child: pid1 = 2868
8, 4, 2, 1
jimmy@jimmy-VirtualBox:~/Documents$ ./hw1 10
parent: pid = 2870
parent: pid1 = 2869
child: pid = 0
child: pid1 = 2870
10, 5, 16, 8, 4, 2, 1
jimmy@jimmy-VirtualBox:~/Documents$ ./hw1 0
Argument is not a positive integer.
jimmy@jimmy-VirtualBox:~/Documents$
```

2. Source code

```
#include <stdio.h>
```

```
#include <unistd.h>
```

```
#include <sys/types.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
pid_t pid, pid1;
```

```
int n;
```

```

if (argc == 1) {
    fprintf(stderr, "Usage: ./a.out <starting value>\n");

    return -1;
}

n = atoi(argv[1]);

/* add your code below, following the code structure of Figure 3.34 (page 152) */

pid=fork();

//check if n is positive
if(n<=0){
    fprintf(stderr, "Argument is not a positive integer.\n");
    return 1;
}

//check if fork fails
if(pid<0){
    fprintf(stderr, "Fork failed.");
    return 1;
}

```

```

}

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

    pid1=getpid();

    printf("child: pid = %d\n",pid);

    printf("child: pid1 = %d\n",pid1);

    while(n>1){

        if(n%2==0){

            printf("%d, ",n);

            n=n/2;

        }

        else{

            printf("%d, ",n);

            n=(3*n)+1;

        }

    }

    printf("%d\n",n);

}

else{ //parent process

    pid1=getpid();

    printf("parent: pid = %d\n",pid);

    printf("parent: pid1 = %d\n",pid1);

    wait(NULL);

}

```

```
return 0;
```

```
}
```