



Εργαστήριο 8

Ασκήσεις: Process Management and Signals

Άσκηση 1

Άσκηση 2

```
// parent function
#include .....

int pid1, pid2;

void Handler_C(int a)
{
    kill(pid1, SIGUSR1);
    kill(pid2, SIGUSR1);
}

int main (int argc, char *argv[])
{
    int ret, status;

    pid1=fork();

    if (pid1==-1)
    {
        printf("Error creating the first child \n");
        exit (1);
    }
    if (pid1==0)
    {
        ret=execl("/home/research/hpollas/epl428/process_mng/compute_pi1",
"compute_p1", argv[1], NULL);
        if (ret==-1)
        {
            printf("Could not convert process one \n");
            exit (1);
        }
    }
    else
    {
        pid2=fork();
        if (pid2==-1)
        {
            printf("Error creating the second child \n");
            exit(1);
        }
        if (pid2==0)
        {
            ret=execl("/home/research/hpollas/epl428/process_mng/compute_pi2",
"compute_p2", argv[1], NULL);
            if (ret==-1)
            {
                printf("Could not convert process one \n");
                exit (1);
            }
        }
    }
}
```



```

        else
        {
            printf ("The program started succesfully\n");
            printf ("Press Ctrl^C to see the current values of pi\n\n");
            signal(SIGINT, Handler_C);
            wait(&status);
            wait(&status);
        }
    }

}

// compute_pi1
#include .....

#define orig_pi
3.14159265358979323846264338327950288419716939937510582097494459230781640
628620899862803482534211706798214808651328
float pi=0;

void alarmHandler(int a)
{
    printf("The difference of pi and Compute_pi1 is %.30f\n", orig_pi-pi);
    exit(1);
}

void Handler_Ch(int a)
{
    printf("The difference of pi and Compute_pi1 is %.30f\n", orig_pi-pi);
}

void compute_pi1(char* seconds)
{
    float counter=1;
    int flag=0;

    signal(SIGINT, SIG_IGN);
    signal(SIGUSR1, Handler_Ch);
    signal(SIGALRM, alarmHandler);

    alarm(atoi(seconds));

    while(1)
    {
        if (flag % 2==0)
            pi= pi+(4/counter);
        else
            pi= pi-(4/counter);
        flag++;
        counter+=2;
        //printf("The difference of pi and Compute_pi1 is %.30f\n", orig_pi-
pi);
    }
}

int main(int argc, char * argv[])
{
    compute_pi1(argv[1]);
}

```