*A system call is expected to be significantly more expensive than a procedure call (provided that both perform very little actual computation). A system call involves the following actions, which do not occur during a simple procedure call, and thus entails a high overhead:*

* *A context switch*
* *A trap to a specific location in the interrupt vector*
* *Control passes to a service routine, which runs in 'monitor' mode*
* *The monitor determines what system call has occurred*
* *Monitor verifies that the parameters passed are correct and legal*

*For your experiment, you should measure the total time for a large number of system/function calls, and then find the average time per call in order to overcome the course resolution of your timing functions. For example, here's a sample code for measuring the time taken for a simple system call and a simple function call:*

*#include <sys/time.h>   
#include <unistd.h>   
#include <assert.h>*

*int foo(){ // write your own procedure call  
  return(10);   
}*

*long nanosec(struct timeval t){ /\* Calculate nanoseconds in a timeval structure \*/   
  return((t.tv\_sec\*1000000+t.tv\_usec)\*1000);   
}*

*main(){   
  int i,j,res;   
  long N\_iterations=1000000; /\* A million iterations \*/   
  float avgTimeSysCall, avgTimeFuncCall;   
  struct timeval t1, t2;*

*/\* Find average time for System call \*/   
  res=gettimeofday(&t1,NULL); assert(res==0);   
  for (i=0;i<N\_iterations; i++){   
    j=getpid();   
  }   
  res=gettimeofday(&t2,NULL);   assert(res==0);   
  avgTimeSysCall = (nanosec(t2) - nanosec(t1))/(N\_iterations\*1.0);*

*/\* Find average time for Function call \*/   
  res=gettimeofday(&t1,NULL);  assert(res==0);   
  for (i=0;i<N\_iterations; i++){   
    j=foo();   
  }   
  res=gettimeofday(&t2,NULL);   assert(res==0);   
  avgTimeFuncCall = (nanosec(t2) - nanosec(t1))/(N\_iterations\*1.0);*

*printf("Average time for System call getpid : %f\n",avgTimeSysCall);   
  printf("Average time for Function call : %f\n",avgTimeFuncCall);   
}*

*Sample output on a linux machine :*

*> gcc -O0 testtime.c -o testtime   
> ./testtime   
Average time for System call getpid : 394.778015   
Average time for Function call : 15.080000*