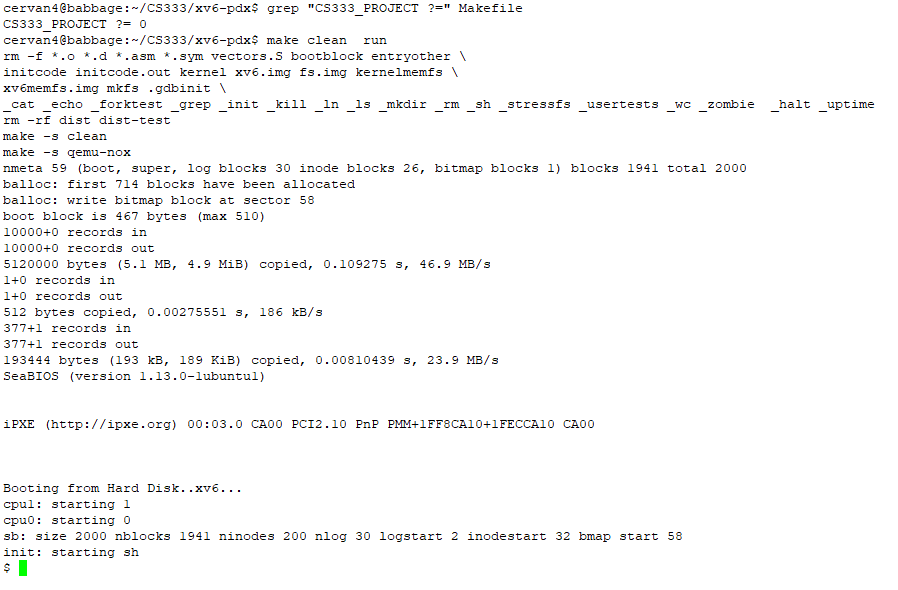
**Test Name:** Correctly compiles and boots with the CS333 P2 macro turned off, with CS333 PROJECT set to 0 in the Makefile

**Test Description:** In this test we will be setting CS333 Project to the value 0 and compline it with no errors

**Expected Results:** Expected results for this test is for it to pass

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything compiled and worked properly.

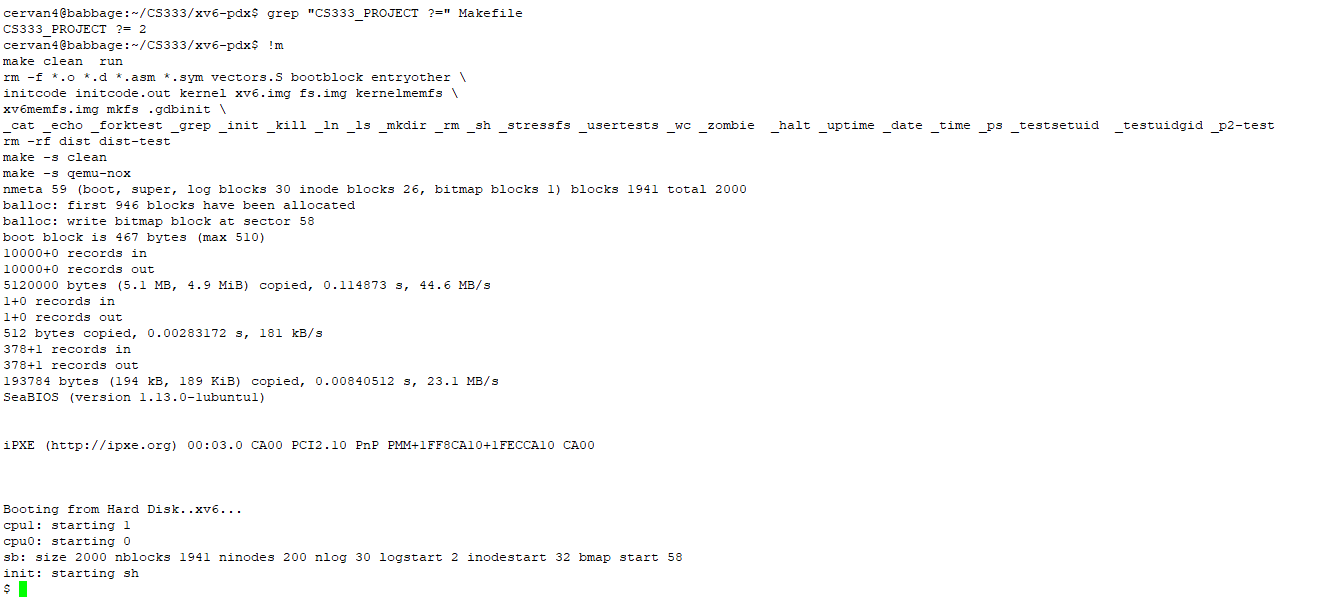
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Correctly compiles and boots with the CS333 P2 macro turned on, with CS333 PROJECT set to 2 in the Makefile

**Test Description:** In this test we will be setting CS333 Project to the value 2 and compline it with no errors

**Expected Results:** Expected results for this test is for it to pass

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything compiled and worked properly.

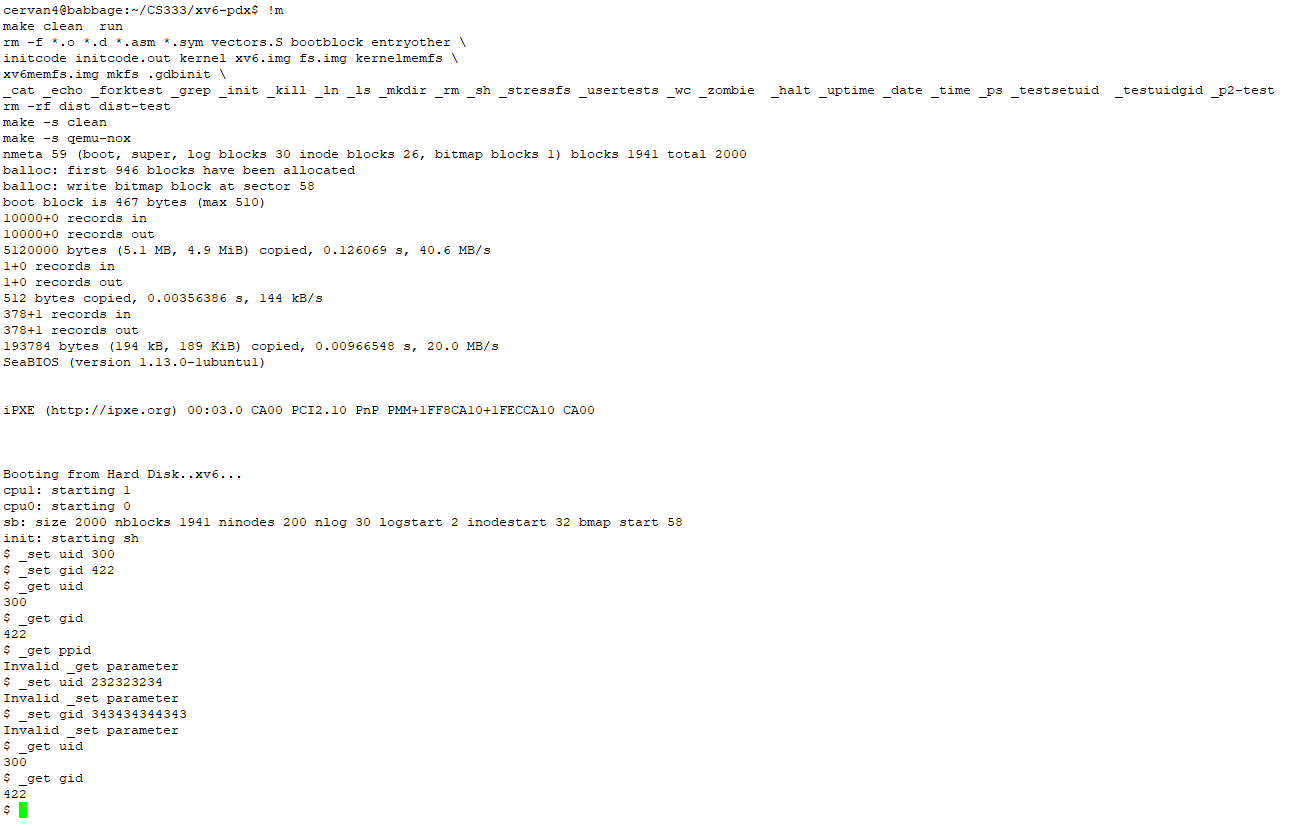
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Correctly set / get UID and GID and get PPID and Correctly handle attempting to set UID and GID to invalid numbers

**Test Description:** In this test we will be setting/getting UID,GID and getting PPID correctly as well we will be attempting to set UID and GID to invalid number

**Expected Results:** Expected results for this test is for it to pass and set the values correct and get the values correct as well get an invalid when trying to set GID or UID to invalid number

**Test Output / Actual Results**

****

**Discussion:** The results were as expected for UID and GID they both set and get the value properly as well wouldn’t allow to set invalid value as expected however getting PPID wasn’t as expected

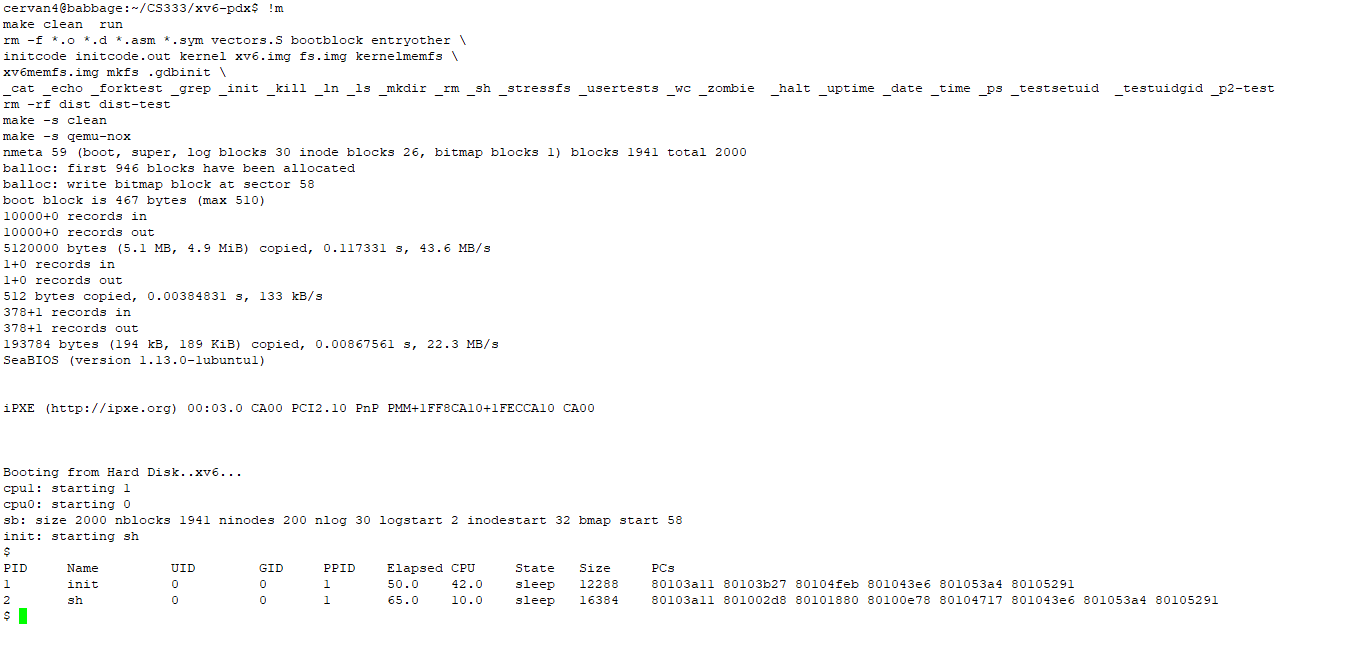
**Indication of PASS/FAIL:** This test UID and GID both passed the tests however get PPID didn’t go as expected

**Test Name:** Show that control – p correctly prints all new information. Fields and headers include: PID Name UID GID PPID Elapsed CPU State Size PCs

**Test Description:** In this test we will show that control -p prints all the new information and all the data is correct.

**Expected Results:** Expected results for this test is for it to pass

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything displayed properly and data was correct

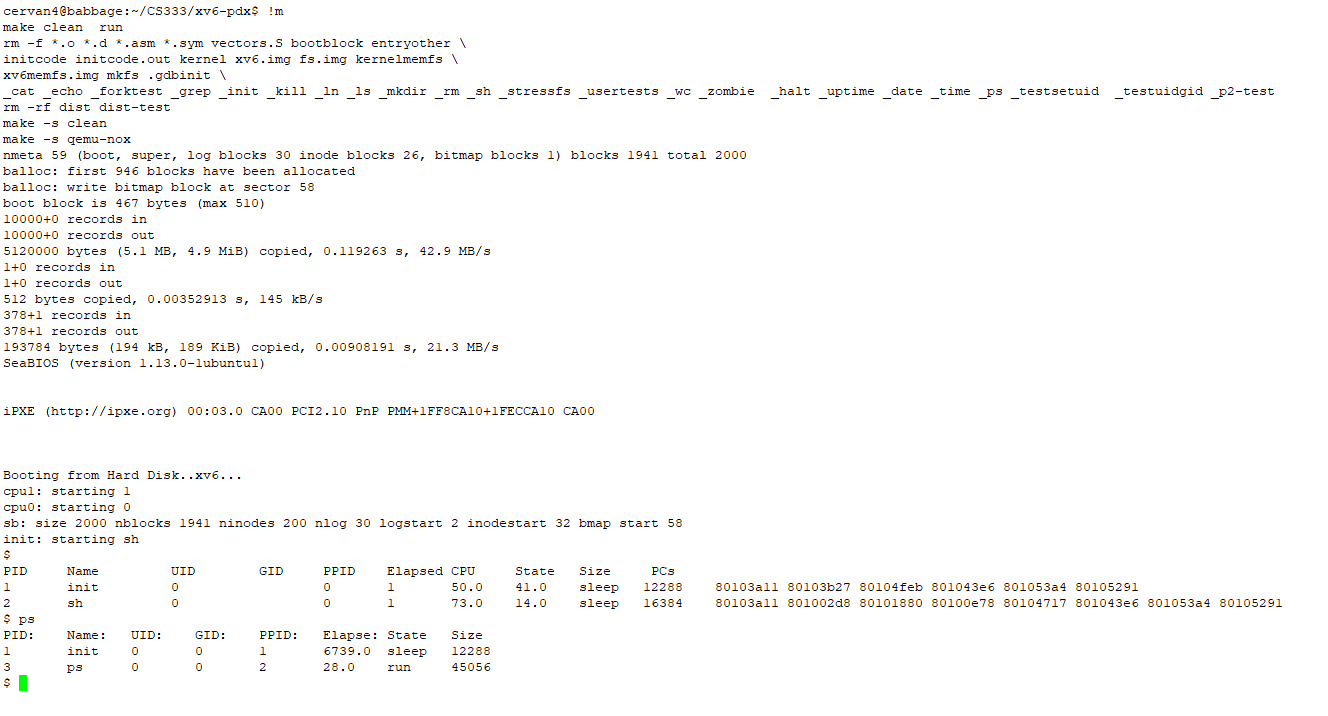
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Test for correct output of ps command. (Compare to control – p)

**Test Description:** In this test we will show the output for ps command and compare to control -p

**Expected Results:** Expected results for this test is for it to pass and print correctly

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything displayed properly and data was correct however from what we can see the data is different in the ps and the control -p commands

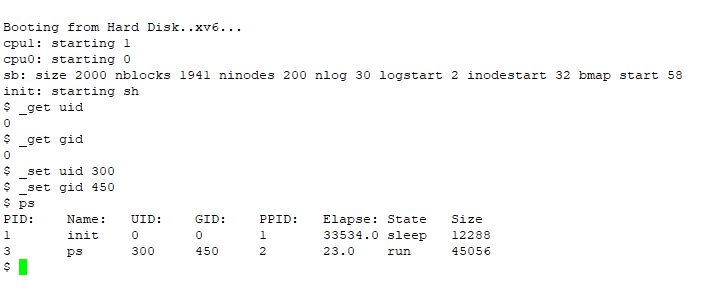
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Test the built-in shell commands to set UID and GID, and show that child processes correctly inherit the new UID and GID values. (ps command can do this)

**Test Description:** In this test we test the built-in shell command to set UID and GID and show that UID and GID inherits the new value

**Expected Results:** Expected results for this test is for it to pass and for everything to inherits correctly.

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything displayed properly and data was correct as well UID and GID inherited correctly so everything was a success

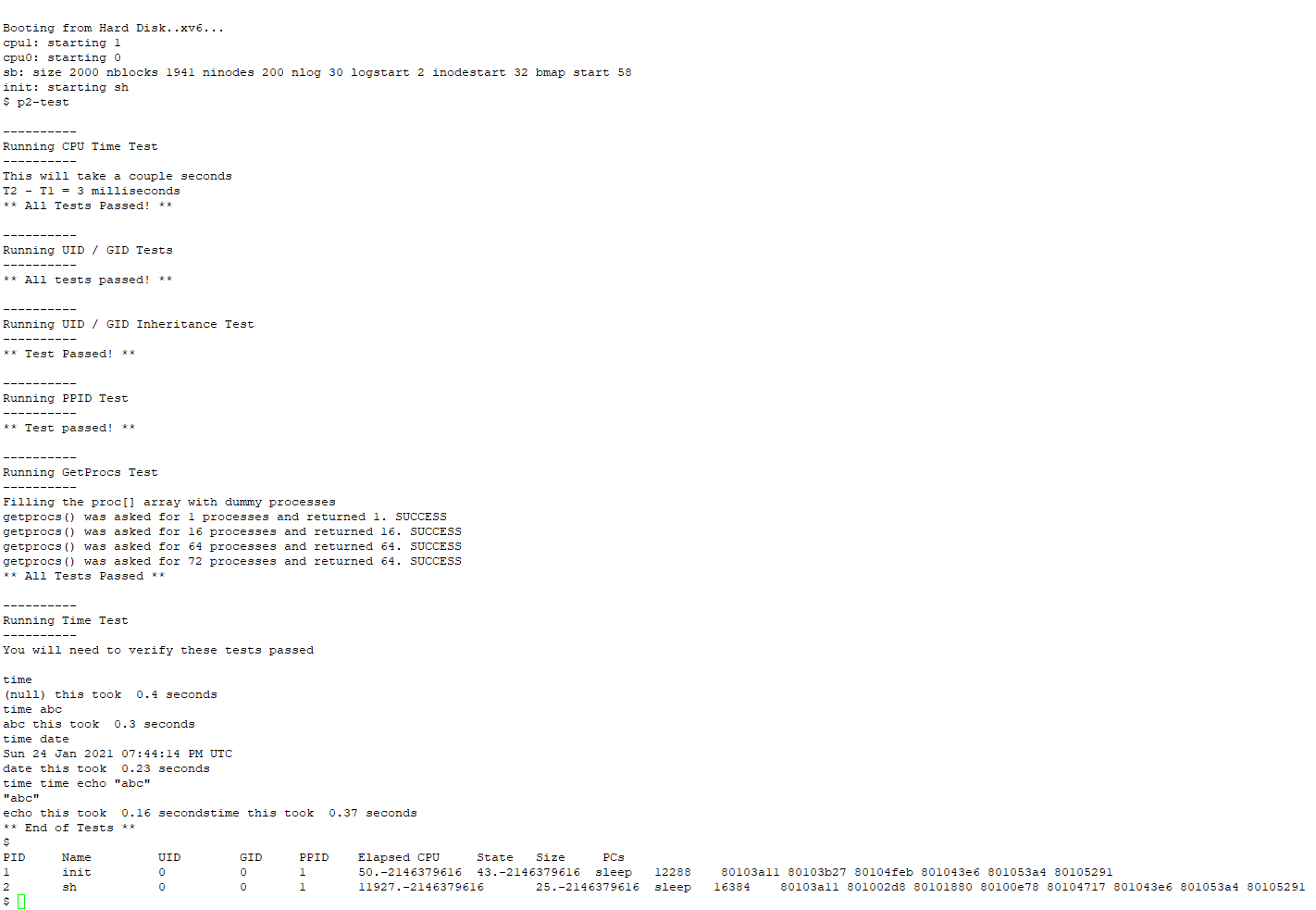
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Tests for getprocs() with 64 active processes. (Staff suggest a test program) – [2 points] Correct output with MAX set to 1, 16, 64, 72, Correctly use control – p as a comparison. (using wait() in your test program will help)

**Test Description:** In this test we test get procs with MAX set to 1,16,64,72 and correctly use control -p as comparison

**Expected Results:** Expected results for this test is for it to pass everything despite the MAX value

**Test Output / Actual Results**

****

**Discussion:** The results were as expected everything passed and as well control -p was used as a comparison

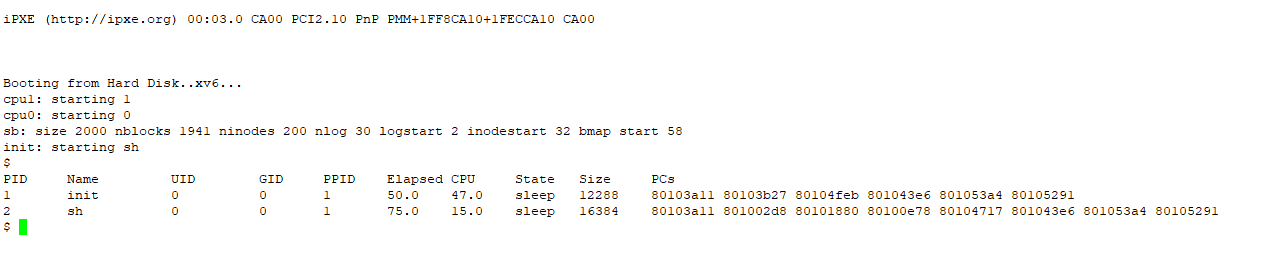
**Indication of PASS/FAIL:** This test Passed

**Test Name:** Test for the correct elapsed CPU time in control – p

**Test Description:** In this test we will use control -p to see if we have the correct elapse time

**Expected Results:** Expected results for this test is for it to give us the correct elapse time

**Test Output / Actual Results**

****

**Discussion:** The results were as expected the elapse time was correct

**Indication of PASS/FAIL:** This test Passed

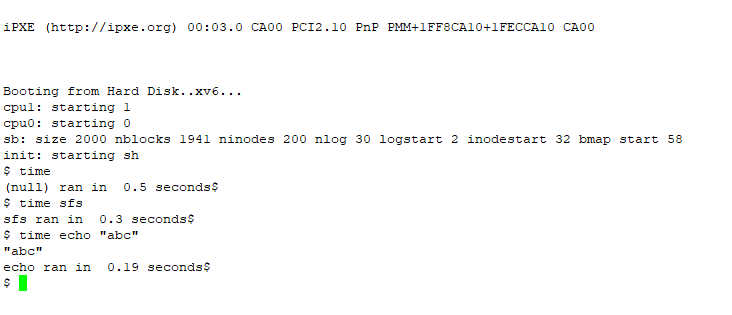
**Test Name:** Tests for the time command:

* Call with no arguments and an invalid argument,
* Call with a valid command argument that takes an argument,
* Show that the calculated time is accurate (control – p elapsed time before and after on a long process would work fine)

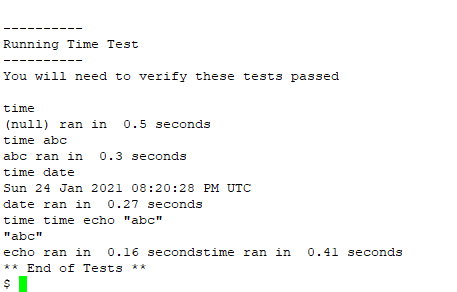
**Test Description:** In this test we will call with no argument and invalid argument as well call with a valid command argument and takes an argument as well show and calculate time is accurate

**Expected Results:** Expected results for this test to pass and for everything to be accurate or close to accurate.

**Test Output / Actual Results**

****

P2-test command

****

**Discussion:** The results were as expected time in comparison to the p2-test command the results were very much close since time echo was 0.16 seconds while mine was 0.19 so very much close so it passed all the tests even when set to nothing

**Indication of PASS/FAIL:** This test Passed