Diego Jared Avina

Dr. Moore

CS 4375-13948 Fall 2023

HW 2: Time Command

Task 1: Implement a time1 command that reports elapsed time

Code: Created time1.c in the user directory, and added the command to UPROGS in Makefile

Results: Running the time1 command yields the elapsed time for a given command execution:

Summary: In Task 1, I implemented a basic time1 command to measure elapsed time. It involved creating a new C file and adding the command to Makefile. The time1 command successfully measures and reports elapsed time for a given task

Difficulties: There were no significant difficulties encountered in Task 1.

Task 2: Keep track of how much CPU time a process has used

Changes: Modified relevant parts of the kernel code to track and update CPU time in the struct proc.

Summary: In Task 2, I made changes to the kernel code to keep track of CPU time for processes. This involved modifying data structures and updating them during process execution. This task was important to accurately report CPU time for processes.

Difficulties: Understanding and locating the relevant parts of the kernel code for this task required careful examination. However, I was able to make the necessary changes successfully.

Task 3: Implement a wait2() system call that waits for a child to exit and returns the child's status and rusage:

Changes: Added a new system call, wwait2(), and modified related kernel files(proc.c, syscall.c, syscall.h, and defs.h)

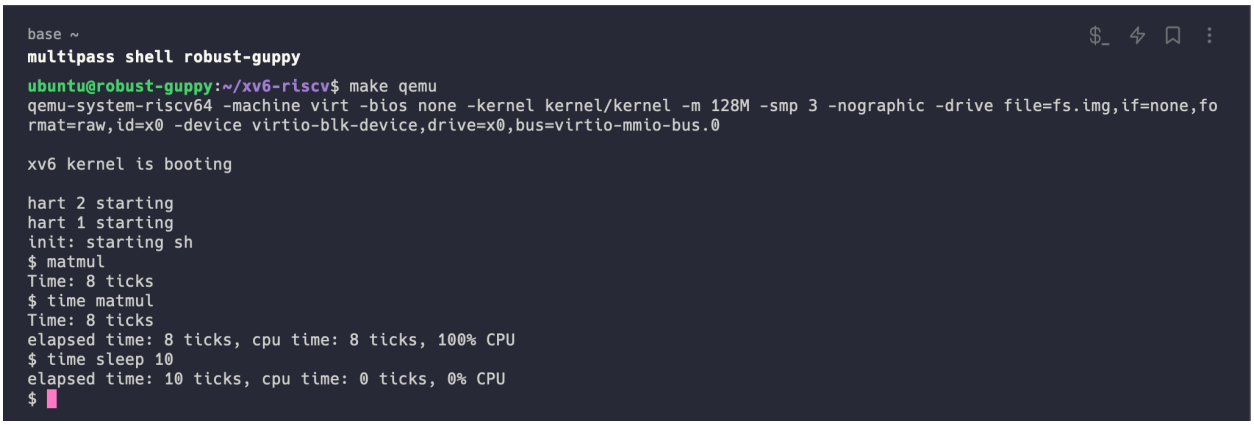
Summary: Task 3 involved implementing a new system call, wait2(), to wait for a child process to exit and return its status and resource usage information. This task required changes to multiple kernel files to integrate the new system call.

Difficulties: The main challenge in this task was ensuring the proper integration of the new system call into the existing kernel code.

Task 4: Implement a time command that runs a command and reports elapsed time, CPU time, and %CPU used:

Code: Created time.c in the user directory, added the command to Makefile, and implemented the time command.

Results: Running the time command successfully reports elapsed time, CPU time, and %CPU usage for a given command



Summary: Task 4 involved implementing the time command, which measures and reports various time-related metrics for a specified command. This task required creating a new user-level command and ensuring it correctly calculates and displays the required metrics.

Difficulties: Task 4 was relatively straightforward since it built upon the knowledge gained from Task 1. However, understanding the internals of time measurement and formatting the output required careful attention to detail.

Extra Credit Task:

Lack of Precision: The time command we implement in xv6 measures time in ticks, which could not be very precise for short-running processes.

No Historical Data: The time command implemented doesn't maintain any history, making it somewhat challenging to analyze how resource usage change