Assignment 1

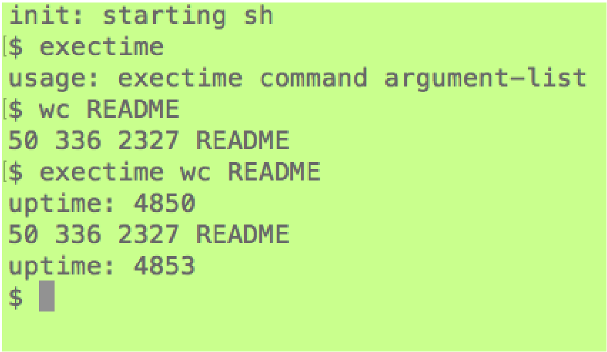
PART A. Using Existing System Calls (2 marks)

Write a user program exectime.c to present the time spent on executing a shell command. Print the start time and completion time in terms of the ticks.

- It should be executed within xv6.
- In xv6, only Makefile can be changed.
- It is ok if the output contains some other information.
- The number of command line arguments for exectime should not be fixed.

Use xv6 system calls: uptime, fork, exec, wait. Pay attention to their syntax (could be different from Unix).

Sample run:



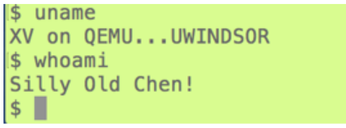
```
init: starting sh
[$ exectime
usage: exectime command argument-list
[$ wc README
50 336 2327 README
[$ exectime wc README
uptime: 4850
50 336 2327 README
uptime: 4853
$ 
```

PART B. Built-in Shell (1 mark)

Implement shell command *whoami* by printing some message to the user. Any printout (e.g., your name) is fine.

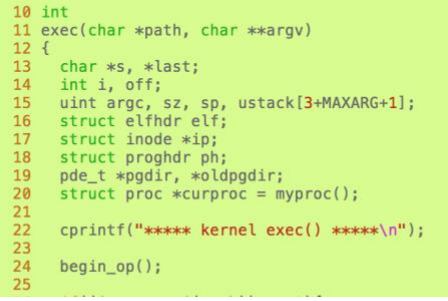
- To be executed in xv6
- In xv6, only sh.c can be modified
- Not required to handle *a composite shell command including whoami*, like *whoami* | *wc*. We will only test command whoami by itself.
- Not allowed to remove or modify the existing implementation of shell commands
- The execution of whoami should not invoke exec().

Sample run:



```
$ uname
XV on QEMU...UWINDSOR
$ whoami
Silly Old Chen!
$
```

Sample testing:

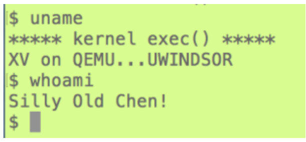


```
int
exec(char *path, char **argv)
{
  char *s, *last;
  int i, off;
  uint argc, sz, sp, ustack[3+MAXARG+1];
  struct elfhdr elf;
  struct inode *ip;
  struct proghdr ph;
  pde_t *pgdir, *oldpgdir;
  struct proc *curproc = myproc();

  cprintf("***** kernel exec() *****\n");

  begin_op();

```



```
$ uname
***** kernel exec() *****
XV on QEMU...UWINDSOR
$ whoami
Silly Old Chen!
$
```

Submission: a zipped file named firstname_lastname.zip containing exectime.c, sh.c, and Makefile.