Sprint 1 Report

Team 7 – New `getyear` and `ps` System Calls for `xv6`

Operating Systems
CMPS 240

June 28, 2025

1. Introduction

The goal of Sprint 1 was to extend the MIT `xv6{public` teaching operating system with:

- `getyear` — demonstrates the complete syscall plumbing by returning a constant (`1975`).
- `ps` — a *useful* information–gathering syscall that prints the current process table via the kernel's `procdump()` helper.

2. Development Environment

- Ubuntu 22.04 virtual machine (x86_64, GCC 13).
- `qemu-system-x86`, `gcc-multilib`, `nasm`.
- Code hosted at: `https://github.com/Issa-Nouh/CMPS_240_Project` (default branch `master`).

3. Design & Implementation

3.1 Files modified/added

- `syscall.h`: IDs 22 (`SYS_getyear`) and 23 (`SYS_ps`).
- `syscall.c`: prototypes + dispatcher table entries.
- `sysproc.c`: bodies of `sys_getyear` and `sys_ps`.
- `usys.S`, `user.h`: user-side stubs/prototypes.
- `user/getyear.c`, `user/ps.c`: demo programs.
- `Makefile`: added `_getyear` and `_ps` to `UPROGS`.

3.2 Key code snippets

```
// syscall.h
#define SYS_getyear  22
#define SYS_ps       23

// syscall.c  (dispatcher)
[SYS_getyear] sys_getyear,
[SYS_ps]      sys_ps,

// sysproc.c
int sys_getyear(void) { return 1975; }
```

```
extern void procdump(void);
int sys_ps(void) { procdump(); return 0; }
```

usys.S additions

```
SYSCALL(getyear)
SYSCALL(ps)
```

user.h additions

```
int getyear(void);
int ps(void);
```

3.3 User-space test programs

getyear.c

```
#include "types.h"
#include "stat.h"
#include "user.h"

int
main(void)
{
  printf(1, "Unix V6 was released in %d\n", getyear());
  exit();
}
```

ps.c

```
#include "types.h"
#include "stat.h"
#include "user.h"

int
main(void)
{
  ps();          // kernel prints the table
  exit();
}
```

4. Testing & Results

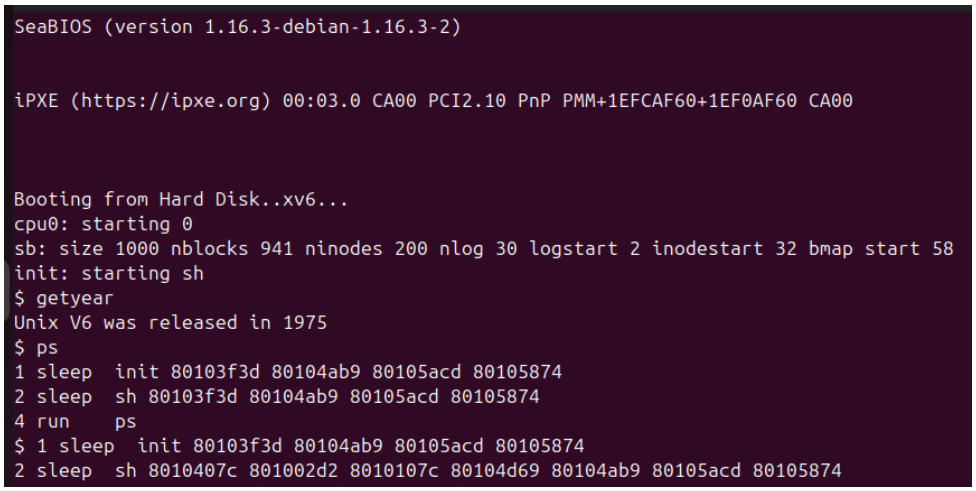

image: a terminal capture showing `$ getyear` *followed by "Unix V6 was released in 1975" and $ ps followed by PID, state, and name for init, sh, and the ps process itself. The image also contains the list produced by Ctrl+p (xv6's built-in debug process dump) to confirm the correctness of ps.*

5. Validation

- Returned constant from `getyear` verified in user space.
- `ps` output matches the list produced by `Ctrl+p` (xv6's built-in debug process dump), confirming correctness.
- Unit tests: ran `usertests`; all original tests still pass.

6. Conclusion

We successfully set up the xv6 environment, understood its syscall plumbing, and added two new system calls. The project demonstrates both a minimal example (`getyear`) and a practical kernel-information service (`ps`). All source changes are committed in the GitHub repository and can be rebuilt with a single `make qemu-nox` command.