RGChanged function and reason:

1) exec.c:

Line 42:

- Change: When loading the program into memory, I change the start sz from 0 to next page.
- Reason: In this way, we do not allocate page table and physical memory to this period of virtual address, result in making the first page unavailable to user.

The changed code is:

sz = PGSIZE ;

2) vm.c:

Line 319:

- Change: In function copyuvm() used for fork(), we change the start of sz to 0x1000 when it copy user data from parent to child.
- Reason: The VA (Virtual address) form 0 to 0xFFF is unavailable, so we should not copy them when forking.

The changed code is:

for(i = PGSIZE; i < sz; i += PGSIZE){

3) syscall.c:

Line 22:

- Change: In function fechint(), we add the code to check whether the address is in 0 to 0xFFF when the process is not the first one.
- Reason: The user program passes an argument in a syscall by argint(), argptr(), argstr(), and in deeper, fetchint() and fetchstr(). So we should make sure the address of the argument (which is attribute addr here) is not in the unavailable VA from 0 to 0xFFF. In addition, the process of pid with 1 is firstly used from system to do the initialization, and we does not make its VA of 0 to 0xFFF unavailable at that time, otherwise the change will result the panic of “init exiting”. Therefore, we should check the pid here to prevent error.

The changed code is:

```
if(addr < PGSIZE && proc->pid != 1)
    return -1;
```

Line 38:

- Change: In function fetchstr(), we do the same in fechint().
- Reason: The same.

The changed code is:

```
if(addr < PGSIZE && proc->pid != 1)
    return -1;
```

Line 65:

- Change: In function argptr(), we add to check whether the returned address “i" is in 0 to 0xFFF when the process is not the first one.

- Reason: The returned i is also the VA for an argument, so it should be checked whether it is in unavailable scope.

The changed code is:

if(i < PGSIZE && proc->pid !=1)
return -1;

4) Makefile:

Line 134:

- Change: We changed the -Ttext 0 to -Ttext 0x1000 in "_%: %.o $(ULIB)".
- Reason: The address is which we set the code/text segment in, in other word, it's the staring of Virtual address for each applications in xv6, so we should not set in the unavailable VA, then we change it to VA of 0x1000 for starting.

The changed code is:

*$(LD) $(LDFLAGS) -N -e main -Ttext 0x1000 -o $@ $^*

Line 171:

- Change: Add new executable file _hw3test1\ and _hw3test2\
- Reason: In order to execute these files (commands) in xv6.

The changed code is:

_hw3test1
_hw3test2

5) hw3test1.c, hw3test2.c

hw3test1.c: This program just try to read the integer form virtual address 0x1000 and 0x0. As we expected, the first one return a value from VA of 0x1000, and the second cause a trap for reading data from unavailable address.

hw3test2.c: This program pass a pointer argument to syscall mkdir(), and this pointer points to virtual address 0x0, so when fetching this string, it will find the addr is less than PGSIZE and return -1. Then the sys_mkdir() in sysfile.c will return -1. Therefore, if it returns -1 but not just run on it, the result is what we want and the program will print "mkdir failed as we expected."

The following is the screen capture of the result of these two tests: