VCC Assignment 3:

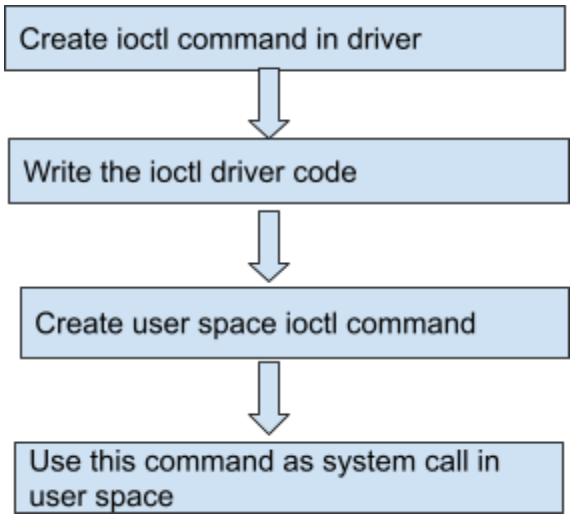
Part 1:

1.a :
The ioctl call :
Kernel space : long ioctl(struct file *fd, unsigned int *request*, unsigned long arg);
User space : int ioctl(int *fd*, unsigned long *request*, ...);
Consists of :

- fd : A valid open file descriptor . So it identifies the device to be controlled.
- request : It is device dependent code which tells the operation to be performed on the device specified by fd. Each request code specifies the argument type whether it is in or out parameter and the size of arguments in bytes.
- The last parameter is …/arg is an untyped pointer to memory. It can pointer to an variable or a data structure required for the operation

1.b :

Kernel space :



2.a:
For guest OS setup ,kvm apis are : KVM_CREATE_VCPU,KVM_GET_VCPU_MMAP_SIZE

2.b:

KVM API	Purpose
KVM_GET_API_VERSION	Check for the version of kvm api. Return 12 for the stable version.
KVM_API_VERSION	Constant value of 12
KVM_CREATE_VM	Create a new VM
KVM_SET_TSS_ADDR	Defines the physical address of a three-page region in the guest physical address space
KVM_SET_USER_MEMORY_REGION	To create, modify or delete a guest physical memory slot
KVM_CREATE_VCPU	Creation of virtual cpu
KVM_GET_VCPU_MMAP_SIZE	Returns the size of the shared memory region between userspace and kvm
KVM_RUN	To run guest vcpu
KVM_TRANSLATE	For virtual address translation. Consists whether the page is valid and if so contains a corresponding offset which needs to be added to the base address to get the physical address.
KVM_GET_REGS	For reading general purpose registers in vcpu
KVM_GET_SREGS	For reading special registers in vcpu
KVM_SET_SREGS	For writing special registers in vcpu
KVM_SET_REGS	For writing general purpose registers in vcpu

a:
For real mode :
No paging is done

For protected mode:
No paging is done

For 32 bit page mode :
There is only 1 level of paging of size 4MBi.e pd directly points to the physical address. The control bits are set using :
pd[0] = PDE32_PRESENT | PDE32_RW | PDE32_USER | PDE32_PS;
Which are for present, read, write , user access bits.
The cr3 register directly points to the base of the pd page table and the segments register are filled up as per the protected mode.

For long mode :
There are 3 page table levels pml4,pdpt,pd and pd entries pointing to the physical address.
The cr3 register is set up to point to the address of plm4. Then the control bits are set :
pml4[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | pdpt_addr;
pdpt[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | pd_addr;
pd[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | PDE64_PS;

The PDE64_PS tells that it is 2M paging, hence the virtual address equals to physical address.
Then the segment registers are setup.

b : 2 MB
As per line `vm_init(&vm, 0x200000);` while initializing the VM it allocates memory using mmap.

c:
The guest page table are setup in :

```
uint64_t pml4_addr = 0x2000;
   uint64_t *pml4 = (void *)(vm->mem + pml4_addr);

   uint64_t pdpt_addr = 0x3000;
   uint64_t *pdpt = (void *)(vm->mem + pdpt_addr);

   uint64_t pd_addr = 0x4000;
   uint64_t *pd = (void *)(vm->mem + pd_addr);
```

```
pml4[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | pdpt_addr;
pdpt[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | pd_addr;
pd[0] = PDE64_PRESENT | PDE64_RW | PDE64_USER | PDE64_PS;
```

d: The PAE stands for physical address extension. It allows addressing more than 4GB memory in 32-bit systems and hence useful in long mode.

4
a:
The guest starts the execution from gva 0x0. It is as per line
regs.rip = 0;

b:
The execution is started at line : ioctl(vcpu->fd, KVM_RUN, 0) and the KVM_RUN api is used to start the guest execution.

5
a:
The values are written/read from a serial port using the out, outb and inb calls. These functions use the asm calls to perform the desired actions.The suffix b indicates byte-width call.

b:
The hypervisor checks if the port is configured for input or output using the kvm_run->io.direction : KVM_EXIT_IO_IN(send data to guest OS) and KVM_EXIT_IO_OUT(data is from guest OS).
Hypervisor accesses the memory location using io.data_offset as offset from vcpu address.

C: May be the value 42 indicates the halting value to the hypervisor.