Computer Science 457 Assignment 2:

Question 1:

A. Direct memory access is a piece of hardware used for transmitting bulk data which is useful for slow devices as it allows the input and output devices to send and receive information without use of CPU.
B. Multiprogramming allows for parallel processing by executing parts of different programs instead of the whole programs one by one which gives the allusion of running programs at the same time.
C. No because removing dma limits the operating systems ability to execute parts of multiple programs concurrently since every part of the program must share and wait for cpu.

Question 2:

A. During the read procedure, the contents of the code are put in a register then a trap to the kernel is called. In the kernel space the system read function is dispatched and a handler is called before returning to the caller. The stack pointer is incremented and the user is given the contents of the system call read.
B. No it does not, the library function is only wrapping the executing of a system call which can have a different name.

Question 3:

A. Yes, a process becomes blocked when it is waiting for an event to occur or resource to be generated by some other process or program. When this is achieved the program usually exists its blocking state and continues running.
B. No the task must be in a running state before it can be blocked.

Question 4:

A context switch is when a process relinquishes their allocation of CPU to give it to another process, saving the current processes state for later completion. It is necessary for multiprogramming. When a context switch occurs, the kernel fist saves the state and register values of the first process, then the kernel loads the state and register values of the new process and executes it. The process resumes from the where it was first stopped.

Question 7:

Strace for the c program:

```
adamberlak@adamberlak-VirtualBox:~/Assign2$ strace -c ./script sh 3
./script.sh : 208
./another.sh : 53
./test.sh : 47
308
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         3           read
  0.00    0.000000           0         4           write
  0.00    0.000000           0         2           open
  0.00    0.000000           0         4           close
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         3         3 access
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           wait4
  0.00    0.000000           0         1           clone
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         5           mmap2
  0.00    0.000000           0        19           stat64
  0.00    0.000000           0         4           fstat64
  0.00    0.000000           0         1           fcntl64
  0.00    0.000000           0         1           set_thread_area
  0.00    0.000000           0         1           pipe2
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                    58         3 total
```

Strace for the sh program:

```
adamberlak@adamberlak-VirtualBox:~/Assign2$ strace -c ./script.sh sh 3
./script.sh 208
./another.sh 53
./test.sh 47
308
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         8           read
  0.00    0.000000           0         8           open
  0.00    0.000000           0        28        10 close
  0.00    0.000000           0         9         2 waitpid
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           getpid
  0.00    0.000000           0         5         5 access
  0.00    0.000000           0         5           pipe
  0.00    0.000000           0        15           brk
  0.00    0.000000           0         1         1 ioctl
  0.00    0.000000           0         1           dup2
  0.00    0.000000           0         1           getppid
  0.00    0.000000           0         1           getpgrp
  0.00    0.000000           0         1           gettimeofday
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           sysinfo
  0.00    0.000000           0         2           sigreturn
  0.00    0.000000           0         7           clone
  0.00    0.000000           0         1           uname
  0.00    0.000000           0         6           mprotect
  0.00    0.000000           0         5           _llseek
  0.00    0.000000           0        12           rt_sigaction
  0.00    0.000000           0        31           rt_sigprocmask
  0.00    0.000000           0         2           ugetrlimit
  0.00    0.000000           0        14           mmap2
  0.00    0.000000           0         2           stat64
  0.00    0.000000           0         7           fstat64
  0.00    0.000000           0         1           getuid32
  0.00    0.000000           0         1           getgid32
  0.00    0.000000           0         1           geteuid32
  0.00    0.000000           0         1           getegid32
  0.00    0.000000           0         3         1 fcntl64
  0.00    0.000000           0         1           set_thread_area
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                   184        19 total
```

Time for the c program:

```
adamberlak@adamberlak-VirtualBox:~/Assign2$ time ./script sh 3
./script.sh : 208
./another.sh : 53
./test.sh : 47
308

real    0m0.003s
user    0m0.000s
sys     0m0.000s
```

Time for the sh program:

```
adamberlak@adamberlak-VirtualBox:~/Assign2$ time ./script.sh sh 3
./script.sh 208
./another.sh 53
./test.sh 47
308

real    0m0.007s
user    0m0.000s
sys     0m0.000s
```

Observation: The time for the sh program is notably longer than the c program. This is because the sh program makes much more system calls/calls than the c program. 19 system calls compared to 3, and 184 calls as opposed to 58, comparing the sh program and c program respectively.