CLAIMS

We claim:

1. A system to manage graphics workloads, comprising:

a graphics hardware component;

a host processor to execute an application lacking system level privileges;

a memory device to issue a notification of a workload submission by the application;

a controller including,

a notification module to receive the notification of the workload submission; and

a scheduler to schedule a transfer of the workload submission to the graphics hardware component for execution.

2. The system of claim 1, wherein the transfer is to bypass an operating system associated with the host processor.

3. The system of claim 1, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

4. The system of claim 1, wherein the notification is to be an interrupt.

5. The system of any one of claims 1 to 4, wherein the scheduler includes:

a retrieval component to retrieve the workload submission from a work queue associated with the application;

a parser to parse the workload submission to obtain one or more commands;

an enqueue component to add the one or more commands to a submission queue associated with the graphics hardware component; and

a dequeue component to transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

6. The system of claim 5, wherein the parser is to modify at least one of the one or more commands, add at least one command to the one or more commands, and prioritize the one or more commands.

7. A method of operating a controller, comprising:

receiving, at the controller, a notification of a workload submission by an application lacking system level privileges; and

using the controller to schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the controller has system level privileges.

8. The method of claim 7, wherein the transfer bypasses an operating system.

9. The method of claim 7, wherein the transfer bypasses a kernel mode driver associated with the graphics hardware component.

10. The method of claim 7, wherein the notification is an interrupt.

11. The method of any one of claims 7 to 10, wherein using the controller to schedule the transfer includes:

retrieving the workload submission from a work queue associated with the application;

parsing the workload submission to obtain one or more commands;

adding the one or more commands to a submission queue associated with the graphics hardware component; and

transferring the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

12. The method of claim 11, wherein parsing the workload submission includes:

modifying at least one of the one or more commands;

adding at least one command to the one or more commands; and

prioritizing the one or more commands.

13. A controller to manage graphics workloads, comprising:

a notification module to receive a notification of a workload submission by an application lacking system level privileges; and

a scheduler to schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the scheduler has system level privileges.

14. The controller of claim 13, wherein the transfer is to bypass an operating system.

15. The controller of claim 13, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

16. The controller of claim 13, wherein the notification is to be an interrupt.

17. The controller of any one of claims 13 to 16, wherein the scheduler includes:

a retrieval component to retrieve the workload submission from a work queue associated with the application;

a parser to parse the workload submission to obtain one or more commands;

an enqueue component to add the one or more commands to a submission queue associated with the graphics hardware component; and

a dequeue component to transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

18. The controller of claim 17, wherein the parser is to modify at least one of the one or more commands, add at least one command to the one or more commands, and prioritize the one or more commands.

19. At least one computer readable storage medium comprising a set of instructions which, when executed by a controller, cause the controller to:

receive a notification of a workload submission by an application lacking system level privileges; and

schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the controller is to have system level privileges.

20. The at least one computer readable storage medium of claim 19, wherein the transfer is to bypass an operating system.

21. The at least one computer readable storage medium of claim 19, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

22. The at least one computer readable storage medium of claim 19, wherein the notification is to be an interrupt.

23. The at least one computer readable storage medium of any one of claims 19 to 22, wherein the instructions, when executed, cause the controller to:

retrieve the workload submission from a work queue associated with the application;

parse the workload submission to obtain one or more commands;

add the one or more commands to a submission queue associated with the graphics hardware component; and

transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

24. The at least one computer readable storage medium of claim 23, wherein the instructions, when executed, cause the controller to:

modify at least one of the one or more commands;

add at least one command to the one or more commands; and

prioritize the one or more commands.

25. A controller to manage graphics workloads, comprising:

means for receiving, at the controller, a notification of a workload submission by an application lacking system level privileges; and

means for using the controller to schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the controller is to have system level privileges.

CLAIMS

We claim:

1. A system comprising:

a graphics hardware component;

a host processor to execute an application lacking system level privileges;

a memory device to issue a notification of a workload submission by the application;

a controller including,

a notification module to receive the notification of the workload submission; and

a scheduler to schedule a transfer of the workload submission to the graphics hardware component for execution.

2. The system of claim 1, wherein the transfer is to bypass an operating system associated with the host processor.

3. The system of claim 1, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

4. The system of claim 1, wherein the notification is to be an interrupt.

5. The system of claim 1, wherein the scheduler includes:

a retrieval component to retrieve the workload submission from a work queue associated with the application;

a parser to parse the workload submission to obtain one or more commands;

an enqueue component to add the one or more commands to a submission queue associated with the graphics hardware component; and

a dequeue component to transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

6. The system of claim 5, wherein the parser is to modify at least one of the one or more commands, add at least one command to the one or more commands, and prioritize the one or more commands.

7. A method of operating a controller, comprising:

receiving, at the controller, a notification of a workload submission by an application lacking system level privileges; and

using the controller to schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the controller has system level privileges.

8. The method of claim 7, wherein the transfer bypasses an operating system.

9. The method of claim 7, wherein the transfer bypasses a kernel mode driver associated with the graphics hardware component.

10. The method of claim 7, wherein the notification is an interrupt.

11. The method of claim 7, wherein using the controller to schedule the transfer includes:

retrieving the workload submission from a work queue associated with the application;

parsing the workload submission to obtain one or more commands;

adding the one or more commands to a submission queue associated with the graphics hardware component; and

transferring the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

12. The method of claim 11, wherein parsing the workload submission includes:

modifying at least one of the one or more commands;

adding at least one command to the one or more commands; and

prioritizing the one or more commands.

13. A controller comprising:

a notification module to receive a notification of a workload submission by an application lacking system level privileges; and

a scheduler to schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the scheduler has system level privileges.

14. The controller of claim 13, wherein the transfer is to bypass an operating system.

15. The controller of claim 13, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

16. The controller of claim 13, wherein the notification is to be an interrupt.

17. The controller of claim 13, wherein the scheduler includes:

a retrieval component to retrieve the workload submission from a work queue associated with the application;

a parser to parse the workload submission to obtain one or more commands;

an enqueue component to add the one or more commands to a submission queue associated with the graphics hardware component; and

a dequeue component to transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

18. The controller of claim 17, wherein the parser is to modify at least one of the one or more commands, add at least one command to the one or more commands, and prioritize the one or more commands.

19. At least one computer readable storage medium comprising a set of instructions which, when executed by a controller, cause the controller to:

receive a notification of a workload submission by an application lacking system level privileges; and

schedule a transfer of the workload submission to a graphics hardware component for execution, wherein the controller is to have system level privileges.

20. The at least one computer readable storage medium of claim 19, wherein the transfer is to bypass an operating system.

21. The at least one computer readable storage medium of claim 19, wherein the transfer is to bypass a kernel mode driver associated with the graphics hardware component.

22. The at least one computer readable storage medium of claim 19, wherein the notification is to be a deferred procedure call.

23. The at least one computer readable storage medium of claim 19, wherein the instructions, when executed, cause the controller to:

retrieve the workload submission from a work queue associated with the application;

parse the workload submission to obtain one or more commands;

add the one or more commands to a submission queue associated with the graphics hardware component; and

transfer the one or more commands from the submission queue to the graphics hardware component in accordance with a graphics submission policy.

24. The at least one computer readable storage medium of claim 23, wherein the instructions, when executed, cause the controller to:

modify at least one of the one or more commands;

add at least one command to the one or more commands; and

prioritize the one or more commands.