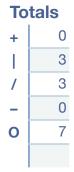
Perform operating system tasks that are typically viewed as "power user" activities.		Outcomes	HW 0124	HW 0205	HW 0219	So Far
shell. Monitor and manage processes and memory use. Navigate, monitor, and manage a logical file system. Redirect input and output streams to and from files, processes, and networked computers. Interact with operating systems across the network. Implement common operating system functionalities and algorithms. Build and deploy an operating system kernel. Define, implement, and invoke a new system call. Write a simple operating system shell. Simulate or implement standalone demonstrations of operating system scenarios and algorithms. Create a virtual disk and navigate it at the byte level. Demonstrate genre literacy within the operating system field. Identify and perform equivalent tasks across distinct operating system platforms. State and describe seminal personalities and milestones from the field's history. Follow academic and technical best practices throughout the course. Write syntactically correct, functional code. Write code that is easily understood by programmers other than yourself. Use available resources and documentation to find required information. 4e Use version control effectively.	1					
1c Navigate, monitor, and manage a logical file system. 1d Redirect input and output streams to and from files, processes, and networked computers. 1e Interact with operating systems across the network. 2 Implement common operating system functionalities and algorithms. 2a Build and deploy an operating system kernel. 2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively.	1a		I	0		0
1d Redirect input and output streams to and from files, processes, and networked computers. 1e Interact with operating systems across the network. 2 Implement common operating system functionalities and algorithms. 2a Build and deploy an operating system kernel. 2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively.	1b	Monitor and manage processes and memory use.		0		0
networked computers. 1e Interact with operating systems across the network. 2 Implement common operating system functionalities and algorithms. 2a Build and deploy an operating system kernel. 2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Follow academic and describe seminal personalities and milestones from the field of the course. 4 Write syntactically correct, functional code. 4 Demonstrate proper separation of concerns. 4 Use available resources and documentation to find required information.	1c	Navigate, monitor, and manage a logical file system.		0		0
2 Implement common operating system functionalities and algorithms. 2a Build and deploy an operating system kernel. 2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Define in plement control effectively. 4 Define, included and well and invoke a new system call. 4 Define in plement standalone demonstrations and invoke a new system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations and invoke a new system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement standalone demonstrations of operating system call. 4 Define in plement	1d			0		0
algorithms. 2a Build and deploy an operating system kernel. 2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 4 Define implement and invoke a new system call. 5 Define implement and invoke a new system call. 5 Define implement and invoke a new system call. 5 Define implement and invoke and algorithms. 6 Define implement and invoke a new system call. 7 Define implement and invoke a new system call. 8 Define implement and system plants. 8 Define implement and system plants. 9 Define implement and system plants. 9 Define implement a	1e	Interact with operating systems across the network.		0		0
2b Define, implement, and invoke a new system call. 2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Demonstrate proper separation of concerns and documentation to find required information.	2					
2c Write a simple operating system shell. 2d Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4 Use version control effectively. 4 Demonstrate proper separation of concerns. 5 Demonstrate proper separation of concerns. 6 Demonstrate proper separation of concerns. 7 Demonstrate proper separation of concerns. 9 Demonst	2 a	Build and deploy an operating system kernel.				
Simulate or implement standalone demonstrations of operating system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4 Use version control effectively. 4 Demonstrate proper separation of concerns. 5 Demonstrate proper separation of concerns. 6 Demonstrate proper separation of concerns. 7 Demonstrate proper separation of concerns.	2 b	Define, implement, and invoke a new system call.				
system scenarios and algorithms. 2e Create a virtual disk and navigate it at the byte level. 3 Demonstrate genre literacy within the operating system field. 3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4 Use version control effectively. 4 O I O	2c	Write a simple operating system shell.				
3a Identify and perform equivalent tasks across distinct operating system platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 I I O 4 I O	2d				/	/
3a Identify and perform equivalent tasks across distinct operating system platforms. 1 1 1 1 1 1 1 1 1	2e	Create a virtual disk and navigate it at the byte level.				
platforms. 3b State and describe seminal personalities and milestones from the field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Follow academic and technical best practices throughout the course. 4 I I I I I I I I I I I I I I I I I I	3	Demonstrate genre literacy within the operating system field.				
field's history. 4 Follow academic and technical best practices throughout the course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Follow academic and technical best practices throughout the course. 4 I / / / / / / / / / / / / / / / / / /	3a				I	I
course. 4a Write syntactically correct, functional code. 4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4 Write syntactically correct, functional code. 4 I I 4 O I O	3b	·				
4b Demonstrate proper separation of concerns. 4c Write code that is easily understood by programmers other than yourself. 4d Use available resources and documentation to find required information. + 0 0 4e Use version control effectively. + 0 0	4	·				
4c Write code that is easily understood by programmers other than yourself. I I 4d Use available resources and documentation to find required information. + O I O 4e Use version control effectively. + O I O	4a	Write syntactically correct, functional code.			/	/
yourself. 4d Use available resources and documentation to find required information. 4e Use version control effectively. 4	4b	Demonstrate proper separation of concerns.			-	
required information. + 0 0 4e Use version control effectively. + 0 0	4c				I	I
	4d		+	0	I	0
4f Meet all designated deadlines. + - /	4e	Use version control effectively.	+	0		0
	4f	Meet all designated deadlines.	+	-		/



Pending: Screenshots noted for a couple of the exercises in HW 0205, but still missing a lot of portions; I held off until there is more to evaluate.