Example of using pipelines to build interesting programs: Donald Knuth vs Doug McIlroy
	Knuth's thousands of lines of code vs McIlroy's 6-command pipeline written in 30 seconds

the exec family of functions are used to run other processes, whether from files (execvp) or from paths (execv)
	the process replaces the process that calls it (they are given the same pid)
	it takes the arguments that it's given and puts them on the stack in a contiguous chunk
	the environment is also put on the stack (there's more on the environment in the book, but it's unimportant for this class)
	the new process also copies all file descriptors
		for example, it inherits standard output from its parent
		if the original process was redirected to a file or piped, the new process inherits this redirection/pipe

How file descriptors work under the hood
	when a process is created, it is given a file descriptor table in kernel space where indexes are the file descriptors and are hooked up to their respective I/O devices
		for example, index 0 (stdin) is hooked up to the keyboard, indexes 1 (stdout) and 2 (stderr) are hooked up to the screen
	when fork is called, a separate copy of the fd table for the child is created in kernel space
		then, when the rest of exec happens and the process is overwritten with the new process, it doesn't change the fd table, so it has the same file descriptors as its parent/caller of exec
	file descriptors can be changed before exec is called in the child, so things you can do before calling exec:
		set up pipes
		set up redirections
	redirections: new system call dup2(fd1,fd2)
		makes fd2 point to fd1 in the file descriptor table
		after using dup2, always close fd1
	dup2 is also used to make pipes
	pipes must be set up outside of the child process so that when the fork happens, both the parent and the child have the pipe in the fd table

A Redirection Example

A Pipe Example

To set up the write end of the pipe:
	dup2(pipefd[1], STDOUT_FILENO);
	close(pipefd[1]);
	close(pipefd[0]);

Example of using a pipeline to combine simple shell programs into an interesting and complex program: the Sieve of Eratosthenes	unhooking and rehooking each pipe between pipesieve and filtermultiples