OS Godz – R1 – Programmer's Manual

Nathan Fielding – Logan Yokum – Dylan Smith – Ethan Boddy

void commhand()

Function to manage and execute commands based on input from the serial port. Takes in no parameters and the return type is void. Function will compare the input from the serial port to each of the standard commands (help, shutdown, date, time, version) and call that command if the input was verified. If the input does not match any valid command then an error message will be displayed and the user will be prompted to enter a valid command.

void help(char* args)

Function to display the list of commands that can be executed. The parameter is a character pointer that will determine which options and instructions are output. If there are no arguments passed into this function then it will print a numbered list of commands in the order of help, shutdown, time, date and version. Any input from the user will be compared to these commands and if a valid command is input then an instructional prompt will be displayed explaining how to use the selected command. If an invalid command is input then a message will be displayed prompting the user to enter a valid command.

int shutdown()

Function to exit the MPX system. No arguments should be passed into it and it will return an integer value indicating whether the shutdown was successful meaning the user typed shutdown twice after being prompted to confirm. A 1 will be returned if unsuccessful (too many arguments passed or no shutdown confirmation) and a 0 will be returned if the shutdown was successful. The input from the user is loaded into a character array, sanitized (trailing whitespace, extra arguments) and compared with the confirmation input before executing.

void time(char* args)

Function to read and write the time from the Real Time Clock. Takes in a character pointer to determine if the time is to be accessed or mutated. If no arguments are passed and a new line "\n" is detected then the current time will be displayed. The return type is void. The hours, minutes and seconds are accessed individually via the bytes from the RTC, converted to integers, and separated by commas before being output. If arguments are passed into this function then the input is parsed by the colon delimiters and each element (hours, minutes, seconds) is stored and checked for validity. If the arguments are valid then the new time is set. If the arguments are invalid then a message will be displayed containing valid input bounds and a prompt to retry.

void date(char* args)

Function to read and write the date from the Real Time Clock. Takes in a character pointer to determine if the date is to be accessed or mutated. If no arguments are passed and a new line "\n" is detected then the current date will be displayed. The return type is void. The month, day and year are accessed individually via the bytes from the RTC, converted to integers, and separated by commas before being output. If arguments are passed into this function then the input is parsed by the colon delimiters and each element (month, day, year) is stored and checked for validity. If the arguments are valid then the new date is set. If the arguments are invalid then a message will be displayed containing valid input bounds and a prompt to retry.

void version()

Function to output the current milestone (R1, R2, etc.) in the MPX development. Takes in no parameters and the return type is void.

int serial_poll(device dev, char* buffer, size_t len)

Function to read input from the serial port. Takes in a device struct to read the input from, a character pointer as the user-provided buffer, and a size_t as the length of the user-provided buffer. If successful then it will return the number of bytes read from the device as an integer. If unsuccessful then it will return a negative integer.

char* gettime()

Helper function used within **time(char* args)** to fetch the current time from the RTC. Does not take in any arguments and will return the current time as character pointer. The hours, minutes and seconds are accessed individually via the bytes from the RTC, converted to integers, and separated by commas before being output.

char* getdate()

Helper function used within **date(char* args)** to fetch the current date from RTC. Does not take in any arguments and will return the current date as a character pointer. The month, day and year are accessed individually via the bytes from the RTC, converted to integers, and separated by commas before being output.

char* itoa(int n)

Function to convert an integer into a null-terminated string. Takes in the integer to be converted and returns the string version of the integer as a character pointer.

int dtoh(int dec)

Function to convert a decimal number to a hexadecimal number. Takes in the decimal integer to be converted and returns the hexadecimal version as an integer.

int htod(int hex)

Function to convert a hexadecimal number to a decimal number. Takes in the hexadecimal integer to be converted and returns the decimal version as an integer.

int validnum(const char* s)

Function to check if a number is valid. Takes in a constant character pointer and checks each index of the string to verify that it is a numerical digit ranging from 0-9. If any of the digits fall out of this range then a 0 will be returned. If all digits are verified to be valid then a 1 is returned.

int println(const char* message)

Function to print a string on its own new line in output. Takes in a constant character pointer containing the string to be output and returns the length of the string as an integer.