OS Project Final Phase

Shivam Pandey (918972220)

Note: Behavior of my OS (Firmware) and Application (Cartridge) is in README.txt

API as System Calls Documentation

Name
System Call(3) – To fetch the current register status of the game controller.

Parameters
N/A

Return Value
A character array of length 8 (a byte/8 bits), representing the current state/value of the register.

Name
System Call(6) – To set the background on a particular layer.

Signature with parameters
void setBackground(int control_index, uint16_t x, uint16_t y, uint8_t z, uint8_t palette_index, int R, int G, int B, int ALPHA)

Return Value
Return 0 if the operation is performed.

Name
System Call(8) – Draw a small sprite

Signature with parameters
void setSmallSprite(int control_index, uint16_t x, uint16_t y, uint8_t w, uint8_t h, uint8_t z, uint8_t palette_index, int R, int G, int B, int ALPHA)

Return Value
Return 0 if the operation is performed.

Name
System Call(9) – Draw a large sprite

Signature with parameters
void setLargeSprite(int control_index, uint16_t x, uint16_t y, uint8_t w, uint8_t h, uint8_t palette_index, int R, int G, int B, int ALPHA)

Return Value
Return 0 if the operation is performed.

Name
System Call(7) – Display a message/text on the console.

Signature with parameters
void display_text_on_console(char* message_to_display, uint32_t start_index)

Return Value
Return 0 if the operation is performed successfully.

Name
System Call(5) – Switch mode between TEXT MODE and GRAPHICS MODE

Signature with parameters
void switch_mode(uint32_t mode)

Return Value
Return 0 if the operation is performed successfully.

Name
System Call(1) – To get the current timer ticks

Parameters - N/A

Return Value
Returns the current amount Timer Ticks (time units elapsed)

Name
System Call(2) – To get the current MODE (Text or Graphics mode)

Parameters - N/A

Return Value
Returns the value of the MODE CONTROL REGISTER

Name
System Call(4) – To fetch the status of the Interrupt pending register

Parameters - N/A

Return Value
Returns the value of the INTERRUPT PENDING REGISTER (before resolving the raised interrupt)

==========X==========