

aOS - Manual de usuario

Requerimientos del sistema

PC que soporte la arquitectura de 64 bits.

6 Mb de memoria RAM.

Teclado.

Instalación

Para poder usar aOS, se cuenta con una imagen de disco en la carpeta *Image* dentro del directorio. Si no se encuentra este archivo se debe ejecutar el comando *make all* en el directorio raíz. Esa imagen puede cargarse en cualquier dispositivo que desee y que permita ser booteado. También se provee un programa dentro de la raíz llamado *run.sh* que ejecuta *qemu* con todos los parámetros necesarios para su funcionamiento. Tenga en cuenta que debe tener instalado el emulador *qemu* en su sistema.

Funcionalidad

aOS brinda un set de comandos con los que el usuario podrá interactuar con el sistema. Este manual tiene la finalidad de dar a conocer dichos comandos y su funcionamiento. Puede ver también una versión simplificada dentro del sistema usando el comando *help*, el cual será descrito primero. A su vez algunos comandos también reciben parámetros y uno de ellos puede ser *-help* el cual hará que se brinde información específica sobre el uso de ese comando dentro del sistema.

help

Muestra en pantalla la lista de comandos disponibles dentro del sistema y una breve descripción de los mismos.

reboot

Reinicia el sistema operativo.

clear

Limpia la pantalla, volviendo el prompt a la primera línea.

about

Brinda información acerca del desarrollador y de la versión del sistema en uso.

time

imprime en la pantalla la hora del sistema en el formato hh:mm:ss.

timestyle

Permite cambiar el formato que el sistema muestra en la esquina superior derecha.

bkg [-help]|blue|black|green|red

Permite cambiar el color de fondo del intérprete de comandos. Los colores estan predefinidos y se deben insertar como parametro.

ssavercolor [-help]|blue|grey|green|red

Define el color del protector de pantalla. Los colores estan predefinidos y se deben insertar como parametro.

ssavertime [-help]| [0-9]+

Permite definir el tiempo que debe transcurrir hasta que se active el protector de pantalla. Se debe introducir un número entero mayor a 100.

aOS - Manual para desarrolladores

aOS está completamente escrito en el lenguaje de programación C y assembler, por lo tanto las herramientas y aplicaciones que se desarrollen para ser ejecutadas deberán ser programadas en esos lenguajes.

Son provistas para todo aquel que desee desarrollar aplicaciones de usuario para aOS, una serie de herramientas, funciones y definiciones, provistas por el kernel para poder hacerlo. La finalidad de este manual es exponer su ubicación y uso.

Son necesarias las siguientes herramientas para desarrollar: *nasm*, *gcc*, *gcc-multilib*, *ld*.

Librería estándar para el usuario

Esta librería provee funciones basicas para el funcionamiento de cualquier programa con una GUI de modo texto.

*void * memset(void * destination, int32_t character, uint64_t length)*

Función: inicializa una zona de memoria con los valores dados.

Parametros:

- destination: zona de memoria a setear.
- character: valor que tomara cada posicion de memoria.
- length: longitud de la zona que se quiere setear de destination.

*void * memcpy(void * destination, const void * source, uint64_t length)*

Copia a una zona de memoria lo que se encuentra en otra.

Parametros:

- destination: zona de memoria destino.
- source: zona de memoria de la cual se copiará.
- length: longitud de la zona de la zona que se quiere copiar.

unsigned char getchar(void);

Función: obtiene un caracter del buffer de teclado. Si el buffer está vacío se devuelve 0.

int scanf(char, ...);*

Función: permite leer un string en busca de algún patrón deseado

Parametros:

- destination: zona de memoria a setear.
- character: valor que tomara cada posicion de memoria.

- length: longitud de la zona de la zona que se quiere setear de destination.

int putchar(unsigned char character);

Función: imprime en pantalla un carácter.

Parametros:

- character: caracter que se desea imprimir.

int printf(char string, ...);*

Función: clon de la librería estándar de C. Permite imprimir en pantalla strings.

Parametros:

- string: cadena de caracteres principal a imprimir. Puede contener los caracteres especiales %s, %d, %x, %c, los cuales imprimiran una cadena, un número entero, un número entero en hexadecimal o la representación ascii de un número.
- Argumentos variables según la cantidad de elementos especiales agregados en la cadena principal.

int puts(char string);*

Función: imprime en pantalla una cadena sin parámetros especiales.

Parametros:

- string: cadena a imprimir.

int atoi(const char num);*

Función: transforma una cadena de caracteres que representa a un número en un número entero int.

Parametros:

- num: cadena que contiene el string que representa a un número.

char itoa(char* string, int num, int base);*

Función: transforma un número entero en una cadena de caracteres según la base especificada. Por ejemplo devuelve la representación en hexadecimal de un número.

Parametros:

- string: puntero a la posicion donde se dejara escrito el número.
- num: entero que se quiere transformar.
- base: la base del sistema de numeración elegida para su transformación.

void reverse(char string);*

Función: da vuelta una cadena de caracteres.

Parametros:

- string: cadena a transformar.

int strlen(char string);*

Función: devuelve la longitud de una cadena.

Parametros:

- string: cadena de caracteres a medir.

int strcmp(const char cs, const char* ct);*

Función: compara dos cadenas de caracteres. Se devuelve 0 si son iguales y distinto de 0 si no lo son.

Parametros:

- cs: primer cadena a comparar.
- ct: segunda cadena a comparar.

int isspace(int c);

Función: chequea si un caracter es un espacio o un salto de línea o un tab.

Parametros:

- c: caracter a chequear.

int isdigit(int c);

Función: chequea si un caracter es un número.

Parametros:

- c: caracter a chequear.

int isxdigit(int c);

Función: chequea si un caracter es un número válido en un sistema de representación hexadecimal.

Parametros:

- c: caracter a chequear.

int isalpha(int c);

Función: chequea si un caracter es un número o una letra.

Parametros:

- c: caracter a chequear.

int isspecial(int c);

Función: chequea si un caracter es un simbolo especial ('!', '@', '#', '\$', '%', '&').

Parametros:

- c: caracter a chequear.

void printerror(char string);*

Función: imprime un mensaje de error en pantalla, en color rojo.

Parametros:

- string: mensaje a imprimir.

void printwarning(char string);*

Función: imprime un mensaje de advertencia en pantalla, en color amarillo.

Parametros:

- string: mensaje a imprimir.

void printalert(char string);*

Función: imprime un mensaje de alerta en pantalla, en color verde.

Parametros:

- string: mensaje a imprimir.

Syscalls, llamadas al sistema

aOS provee una serie de funciones especiales que son llamadas mediante el uso de la función `_syscall`. Dicha función realiza una interrupción 0x80 al sistema solicitando la operación requerida, similar a lo que sucede en Linux.

void _syscall(int rax, int rbx, uint64_t rcx, int rdx);

A continuación se incluye una tabla con los parámetros posibles y cuál es el resultado obtenido. En los casos que algunos parámetros no sean requeridos por el sistema se puede utilizar cualquier valor, en la tabla se verá el simbolo “-” en dichos casos.

rax	rbx	rcx	rdx	Función
3	1	Puntero a un vector donde se guardará el resultado	Cantidad a leer	Lee caracteres del buffer de teclado
3	3	Puntero a un vector de chars donde se guardará la hora leída	-	Lee la hora del sistema en hh:mm:ss

4	1	Puntero al vector de chars a imprimir.	Cantidad a escribir.	Imprime en pantalla una cadena.
4	2	-	-	Limpia la pantalla
4	3	Color (representado por un char)	-	Setea el color del fondo de pantalla
4	4	-	-	Mueve el cursor a la izquierda
4	5	-	-	Mueve el cursor a la derecha
4	6	-	-	Hace un scroll down
404	-	-	-	Resetea el sistema