yhOS documentation

aceinetx (c) 2022-2025

System calls

id (%eax)	%ebx	%ecx	%edx	description
SYS_WRITEC - 0	character	-	-	Prints a character to the screen
SYS_WRITE - 1	buffer	-	-	Prints a null terminated string to the screen
SYS_GETS - 2	buffer	length	-	Requests a string from the user
SYS_VFSWRITE - 3	filename (char*)	buffer (char*)	buffer_size (dword)	Writes a file to the virtual file system. If no file is found it will create it (if there is enough space). Returns 0 in eax if fails. Returns a content address if doesn't fail
SYS_VFSREAD - 4	filename (char*)	buffer (char*)	buffer_size (dword)	Reads a file from the virtual file system. Returns 0 in eax if fails. Returns a content address if doesn't fail
SYS_VFSQUERY - 5	filename (char*)	-	-	Queries file size from the virtual file system. Returns -1 in eax if fails
SYS_ALLOC	size	-	-	Allocates dynamic memory. (Exposes yalloc to raw assembly programs). Returns allocated address in eax
SYS_FREE	addr	-	-	Free's dynamic memory. (Exposes yfree to raw assembly programs)
SYS_VFSBASE	-	-	-	Returns a pointer to the beggining of virtual file system
SYS_EXEARG	-	-	-	Get next argument passed to the executable from the sheel

A writec system call example in flat assembler:

```
mov eax, SYS_WRITEC
mov ebx, 0x45
int 0x80
```

And in C:

```
syscall(SYS_WRITEC, ‘E’); //Requires
syscall.h
```

Virtual file system

yhOS has a mechanism called "Virtual file system". You can think of it like a virtual hard drive, it stores every file in RAM which unloads on restart.

Virtual file system is structured as a pointer to array of structures called vfs_file. This structure contains these variables: a char* filename, char* content, and dword size. Virtual file system is resizable at runtime via the dynamic memory allocator.

System calls that start with SYS_VFS are not for I/O but for the virtual file system