

NAMA:		PARAF
NPM:		
MODUL:	D1. SYSTEM CALL	
PERIODE:		
ASISTEN:	AD AS BU BZ ID JP KB	
REFERENSI:	Implementing a System Call on Linux 2.6 for i386	

NO	KEMAMPUAN	SELESAI
1	Instalasi	
*	Unduh berkas Virtual Box (ovf dan vmdk) SYSPROG_D_<VERSI> Modul D dari projects.ui.ac.id	
*	Impor dan jalankan Virtual Host	
*	create user "m<NPM>" dan masukkan kedalam daftar sudo	
*	Login dengan "m<NPM>" kemudian check out SVN ke direktori WORK	
*	Membuat laporan-YYMMDD.txt dan script-YYMMDD.txt dan menjalankan script script-YYMMDD.txt pada <SVN>	
2	Manual System Call	
*	Unduh dokumen 'Implement Sys Call' di https://projects.ui.ac.id/attachments/download/292/Implement-Sys-Call-Linux-2.6-i386.pdf	
*	Pada Host, membaca dan memahami dokumen 'Implement Sys Call' sembari melakukan hal dibawah *)	
3	Persiapan	
*	Linux Kernel telah tersedia di /usr/src/[kernel-source]	
*	Masuk ke direktori source code kernel	
*	Ganti ".long sys_coba" menjadi ".long sys_kaliduacall" urutan 327 didalam file <i>arch/x86/kernel/syscall_table_32.S</i>	
*	Ganti "#define __NR_coba" manjadi "#define __NR_kaliduacall" urutan 327 didalam daftar yang terdapat dalam file <i>include/asm-x86/unistd_32.h</i>	
*	Tambahkan "asmlinkage long sys_kaliduacall(int i);" pada akhir baris pada file <i>include/linux/syscalls.h</i>	
*	Tambahkan string " kaliduacall/" pada kalimat "core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/" pada file <i>Makefile</i>	
*	Buat direktori "kaliduacall" pada /usr/src/[kernel-source] dan masuk kedalam	
	buat file <i>kaliduacall.c</i> yang berisi:	
*	<pre>/*---Start of kaliduacall.c-----*/ #include<linux/linkage.h> asmlinkage long sys_kaliduacall(int i) { return i*2; } /*---End of kaliduacall.c-----*/</pre>	
	buat file <i>Makefile</i> yang berisi:	
*	<pre>#####Makefile Start##### obj-y := kaliduacall.o #####Makefile End#####</pre>	
	Lewati 3 langkah ini apabila anda sudah mempunyai config yang optimal	
*	Menyalin konfigurasi kernel existing dari /boot/config-2.6.26-2-686 ke /usr/src/[kernel-source]/.config	
*	<i>Courtesy of Steven Rostedt</i> , untuk mendapatkan compile time yang cukup kecil lakukan: "sudo sh -c `./streamline_config.pl arch/x86/Kconfig > .config-new`" pada /usr/src/[kernel-source]	
*	Ubah <i>.config</i> menjadi <i>.config-original</i> dan ubah <i>.config-new</i> menjadi <i>.config</i>	
4	Konfigurasi	
*	Masukkan konfigurasi kernel dengan perintah "make oldconfig"	
*	Konfigurasi kernel dengan menuconfig dengan perintah "make menuconfig"	
*	Pada "General Setup" => "Local version" tambahkan "- [Nama_Anda]-syscall" tanpa spasi dan tanda kurung.	
*	Pastikan Intel PIIX_IDE terpilih dengan mode M pada "Device Driver" => "ATA/ATAPI/MFM/RLI support" => "Enhanced IDE/MFM/RLI disk/cdrom/tape/ floppy support" => "Intel PIIX/ICH chipsets support", "Generic PCI IDE Chipset Support", dan "Include IDE/ATA-2 DISK support"	

	*	Pastikan Intel PIIX4 terpilih dengan mode M "Device Driver" => "Serial ATA (prod) and Parallel ATA (experimental) drivers" => "ATA SFF support" => "Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA support" dan "Generic ATA support"	
	*	Tanda * berarti driver akan dimasukkan kedalam kernel sedangkan tanda M berarti driver akan dimasukkan ke mode modul kernel dan ubah seperlunya *)	
	*	Simpan ke dalam file <code>.config</code> untuk mempermudah konfigurasi berikutnya	
5	Kompilasi & Instalasi		
	*	Melakukan Kompilasi Kernel dengan perintah "sudo make", dan sementara menunggu selesai dapat membaca-baca <i>Implementing a System Call on Linux 2.6 for i386</i>	
	*	Memasang modul dengan perintah "sudo make modules"	
	*	Install modul dengan perintah "sudo make modules_install"	
	*	Memasang initramfs pada grub initrd dengan menggunakan "sudo mkinitramfs -vo /boot/initrd.img-2.6.26-[Nama]-syscall 2.6.26-[Nama]-syscall"	
	*	Kompresi kernel dengan algoritma bz dengan perintah "sudo make bzImage"	
	*	Pasang kernel ke sistem dengan perintah "sudo make install"	
	*	Melakukan pembaruan daftar boot pada grub dengan perintah "sudo update-grub"	
	*	Nyala-ulangkan Virtual Host ke kernel yang baru saja dikompilasi dengan me-reboot OS	
	*	Login dengan user m<NPM> dan melanjutkan script dengan perintah "script -a script-YYMMDD.txt "	
6	Testing		
		Masuk <SVN> kemudian membuat file <code>testduakallicall.c</code> :	
	*	<pre>#include <unistd.h> #include <stdio.h> #define __NR_kaliduacall 327 long kaliduacall(int i) { return syscall(__NR_kaliduacall, i); } int main() { printf("%d\n", kaliduacall(33)); return 0; }</pre>	
	*	Compile kemudian jalankan tes program tersebut dalam user-space	
7	Penyelesaian		
	*	exit dari script	
	*	Copy image kernel & initramfs yang barusan Anda compile ke <SVN>, kemudian commit <SVN>	
	*	Melakukan sudo halt di Virtual Host	
	*	Mengekspor hasil pekerjaan ke folder D:\Output-[YYMMDD]	

Catatan:

- *) Tidak ada praktek langsung, hanya perlu dicari *resource*-nya, dibaca & dipahami
- <HOME> adalah home direktori Anda, cth: user m150 direktorinya di `/home/m150/`
- <SVN> adalah struktur direktori subversion yang berhubungan dengan modul ini, cth: user m150, direktori <SVN> untuk modul C1 ada di `<HOME>/WORK/C1/`
- <SANDBOX> adalah direktori oprek-oprek yang diletakkan di `<HOME>/sandbox/`

NAMA:		PARAF
NPM:		
MODUL:	D2. MODULE & PROCFS	
PERIODE:		
ASISTEN:	AD AS BU BZ ID JP KB	
REFERENSI:	http://ldp.org/LDP/lkmpg/2.6/html/lkmpg.html	

NO	KEMAMPUAN	SELESAI
1	Instalasi	
*	Melakukan impor berkas image dari D1 ke dalam Virtual Box	
*	Menjalankan Virtual Host yang sudah diimpor.	
*	Jalankan kernel 2.6.26-2-686	
*	Membuat laporan-YYMMDD.txt dan script-YYMMDD.txt dan menjalankan script script-YYMMDD.txt pada <SVN>	

2	Manual Module dan Procfs	
*	Unduh dokumen 'Procfs-Guide' di https://projects.ui.ac.id/attachments/download/312/procfs-guide.pdf	
*	Pada Host, membaca dan memahami dokumen 'Implement Sys Call', beserta halaman referensi sembari melakukan hal dibawah *)	

3	Persiapan & Kompilasi	
*	Lakukan "sudo apt-get install linux-headers-2.6.26-2-686" untuk dapat mengkompilasi Module dan Procfs	
*	Unduh dokumen <i>HelloWorld.c</i> kedalam [SVN]/HelloWorld	
*	buat file <i>Makefile</i> di dalam [SVN]/HelloWorld yang berisi: obj-m += HelloWorld.o	
*	all: make -C /lib/modules/\$(shell uname -r)/build M=\$(PWD) modules	
*	clean: make -C /lib/modules/\$(shell uname -r)/build M=\$(PWD) clean	
*	Jalankan "make" pada [SVN]/HelloWorld	
*	Unduh dokumen <i>kalidua.c</i> kedalam [SVN]/kalidua https://projects.ui.ac.id/attachments/download/313/kalidua.c	
*	buat file <i>Makefile</i> di dalam [SVN]/kalidua dan mengganti HelloWorld.o menjadi kalidua.o	
*	Jalankan "make" pada [SVN]/kalidua	
*	Unduh dokumen <i>procfs_example.c</i> kedalam [SVN]/procfs_example https://projects.ui.ac.id/attachments/download/315/procfs_example.c	
*	buat file <i>Makefile</i> di dalam [SVN]/procfs_example	
*	Jalankan "make" pada [SVN]/procfs_example	

4	Program dan Pemahaman	
*	Jalankan "dmesg tail" untuk mendapatkan pesan pada kernel	
*	Jalankan pada [SVN] "sudo insmod HelloWorld/HelloWorld.ko"	
*	Jalankan "dmesg tail" dan amati baris yang paling baru	
*	Jalankan "lsmod" dan amati baris yang paling atas terdapat modul HelloWorld	
*	Jalankan "ls -al /proc/HelloWorld" dan amati yang terjadi	
*	Jalankan "sudo rmmmod HelloWorld"	
*	Jalankan "dmesg tail" dan amati baris yang paling baru	
*	Pahami kode <i>HelloWorld.c</i>	
*	Jalankan pada [SVN] "sudo insmod kalidua/kalidua.ko"	
*	Jalankan "dmesg tail" dan amati baris yang paling baru	
*	Jalankan "lsmod" dan amati modul kalidua	
*	Jalankan "ls -al /proc/kalidua" dan amati yang terjadi	
*	Jalankan "cat /proc/kalidua" dan amati pesan yang dikeluarkan	
*	Jalankan "dmesg tail" dan amati baris yang paling baru	
*	Jalankan "sudo sh -c "echo 33 > /proc/kalidua""	

*	Jalankan “dmesg tail” dan amati baris yang paling baru	
*	Jalankan “cat /proc/kalidua” dan amati pesan yang dikeluarkan	
*	Jalankan “dmesg tail” dan amati baris yang paling baru	
*	Jalankan pada [SVN] “sudo insmod procfs_example/procfs_example.ko”	
*	Jalankan “dmesg tail” dan amati baris yang paling baru	
*	Jalankan “lsmod” dan amati modul procfs_example	
*	Jalankan “ls -al /proc/procfs_example/” dan amati yang terjadi	
*	Jalankan “cat /proc/procfs_example/bar” dan amati yang terjadi	
*	Jalankan “cat /proc/procfs_example/foo” dan amati yang terjadi	
*	Jalankan “cat /proc/procfs_example/jiffies” dan amati yang terjadi	
*	Jalankan “cat /proc/procfs_example/jiffies_too” dan amati yang terjadi	
*	Jalankan “sudo sh -c “echo foo > /proc/procfs_example/bar””	
*	Jalankan “sudo sh -c “echo bar > /proc/procfs_example/foo””	
*	Jalankan “cat /proc/procfs_example/*” dan amati yang terjadi	
*	Pahami kode <i>kalidua.c</i> dan <i>procfs_example.c</i>	

5	Penyelesaian	
*	exit dari script	
*	tambahkan semua <SVN>, kemudian commit <SVN>	
*	Melakukan sudo halt di Virtual Host	
*	Mengekspor hasil pekerjaan ke folder D:\Output-[YYMMDD]	

Catatan:

- *) Tidak ada praktek langsung, hanya perlu dicari *resource*-nya, dibaca & dipahami
- <HOME> adalah home directory Anda, cth: user m150 direktorinya di /home/m150/
- <SVN> adalah struktur direktori subversion yang berhubungan dengan modul ini, cth: user m150, direktori <SVN> untuk modul c1 ada di <HOME> / WORK/C1/
- <SANDBOX> adalah direktori oprek-oprek yang diletakkan di <HOME>/sandbox/