SP - Minix Kernel Changes

1. **Initial Step**

* Download the minix – dev snapshots [(http://download.minix3.org/iso/snapshot/minix3\_2\_1\_ide\_20131118\_f24634c.iso.bz2)]((http:/download.minix3.org/iso/snapshot/minix3_2_1_ide_20131118_f24634c.iso.bz2))
* Install Minix in VBox
* After Installation, shutdown and remove iso image file from VMachine Settings -> Storage
* Run following commands

setup pkgin update (When simple pkgin don't work)

pkgin in clang-3.4nb1 (Download C Compiler Clang)

Note: Check by clang -v

pkgin in binutils (Download Linker)

write a c program (e.g., hello world)

use "vi hello.c" to create file.

vi works similar to vim. Now, add the lines

of code and compile and link as: clang hello.c -o hello

and execute by ./hello

* Adding all packages
* # pkgin up
* # pkgin\_sets

1. **Adding System Call**

Now first of all we have to check that we have the minix kernel code or not. If /usr/src directory exists, then the code exists in it. Otherwise go to /usr directory and run the following commands:

# cd /usr

# git clone git://git.minix3.org/minix src

The procedure to rebuild kernel after a Git checkout is as follows:

# cd /usr/src

# make build

**Steps**

1. **Adding the PM server table entry**

Open file: vi /usr/src/servers/pm/table.c

Replace the line:

no\_sys, /\* 69 = unused \*/

with:

do\_mysyscall, /\* 69 = mysyscall \*/

### Defining the system call number

### Open file: vi /usr/src/include/minix/callnr.h

### Add this line after 68

#define MYSYSCALL 69

### Between 69 and 71 any number can be used

### Defining the system call's prototype

### Open File: vi /usr/src/servers/pm/proto.h

### Add this line before do\_getsetpriority(void)

int do\_mysyscall(void);

### Implementing the system call

### Open File: vi /usr/src/servers/pm/misc.c

### At the end, write following code

### int do\_mysyscall() {

### printf(“My System Call Invoked”);

### return 0;

### }

### Compiling the system call

You compile the system call and include it in the boot image by completing the following steps:

1. In /usr/src/releasetools type:  
   **make services**
2. Type:   
   **make install**
3. **shutdown** Minix and restart it to run the new version

### Testing the system call

Having compiled the new system call and rebooted, you can use the following program to test the new system call:

#include <lib.h> // provides \_syscall and message

#include <stdio.h>

int main(void) {

// Minix uses message to pass params to a system call

message m;

\_syscall(PM\_PROC\_NR, MYSYSCALL, &m);

/\*

\_syscall leads to the system server identified by PM\_PRO\_NR (the PM server process) invoking the function identified by call number MYSYSCALL with parameters in the message copied to address &m

\*/

return 0;

}