

システムコールの追加方法について

2015/3/30

藤田将輝

1 はじめに

本資料では，新 B4 課題の「(課題 3) IPI を送信するシステムコールの実装と実装手順書の作成」におけるシステムコールの追加手順を示すため，Linux3.17 におけるシステムコールの追加手順について記述する．

2 追加手順

2.1 システムコール本体を用意

システムコールの本体を `arch/x86/kernel` 以下に配置する．システムコール本体のソースコードの例を以下に示す．

2.2 システムコールテーブルへの追加

新たに作成するシステムコールの名前を `arch/x86/syscalls/syscall_64.tbl` に追加する．`arch/x86/syscalls/syscall_64.tbl` を以下のように変更する．

310	64	<code>process_vm_readv</code>	<code>sys_process_vm_readv</code>
311	64	<code>process_vm_writev</code>	<code>sys_process_vm_writev</code>
312	common	<code>kcmp</code>	<code>sys_kcmp</code>
313	common	<code>finit_module</code>	<code>sys_finit_module</code>
+314	common	<code>sys_my_syscall</code>	<code>sys_my_syscall</code>

2.3 プロタイプ宣言を追加

新たに作成するシステムコールのプロトタイプ宣言を追加する．`include/linux/syscalls.h` を以下のように変更する．

```
asmlinkage long sys_kcmp(pid_t pid1, pid_t pid2, int type,
                          unsigned long idx1, unsigned long idx2);
asmlinkage long sys_finit_module(int fd, const char __user *uargs, int flags);
+asmlinkage long sys_my_syscall( char __user *buf, int count );
```

2.4 Makefile の編集

Makefile を編集し、新たに作成するシステムコールを make 対象にする．arch/x86/kernel を以下のように変更する．

```
obj-$(CONFIG_PMC_ATOM)      += pmc_atom.o
+obj-y                      += my_syscall.o
```

2.5 カーネルのビルド

資料 <271-02> を参考にカーネルのビルドを行う．

3 動作確認

3.1 追加したシステムコールを呼び出すプログラムの作成

新たに追加したシステムコールを呼び出すプログラムを任意のディレクトリに作成する．以下に，ソースコードの例を示す．

```
#include<stdio.h>
#include<unistd.h>
#include<sys/syscall.h>

int main(void)
{
    syscall(314);
    return 0;
}
```

3.2 プログラムの実行

作成したプログラムをコンパイルし、実行する．

4 おわりに

本資料では、Linux3.17 におけるシステムコールの追加手順について述べた．