■ Linxu のしくみ

積読になってしまいそうなので，ざっと試しながら一通り読了.
Go, Python, Bash スクリプトを使って分かりやすい例をうまく作ってるなあ, と感心させられた.

内容は知っていたような，分かっていなかったような，っていうような感じ.
sar を使ってあれこれ調べるのは面白かった.
Linux 上の何かを測定する時にひっぱり出すといい，かな.

シグナルハンドラ

p.36 あたり，シグナルハンドラの話で SIGKILL は挙動を変更できない，を試す.

```
#!/usr/bin/python3
import signal
# try to ignore SIGKILL signal
# 1st arg is target signal to assign handler
# 2nd arg is signal handler
signal.signal(signal.SIGKILL, signal.SIG_IGN)
while True:
  pass
```

実行すると,

```
Traceback (most recent call last):
  File "killignore.py", line 9, in <module>
    signal.signal(signal.SIGKILL, signal.SIG_IGN)
  File "/usr/lib/python3.8/signal.py", line 47, in signal
    handler = _signal.signal(_enum_to_int(signalnum), _enum_to_int(handler))
OSError: [Errno 22] Invalid argument
```

なるほど.

スケジューラ

p.55 あたり.

```
sysctl kernel.sched_latency_ns
```

で，レイテンシターゲットの設定がみえると書いてあるが,
手元の環境 (Ubuntu 20.04.5, Linux 5.15.0-48-generic) では見えない.
で調べると，https://forum.endeavouros.com/t/sysctl-output-changed-from-kernel-5-10-to-5-13-why/17097 に解が.

```
sudo cat /sys/kernel/debug/sched/latency_ns
```

として 24000000 が見えた.

メモリ管理システム

p.81 図 04-10 の 物理メモリ側は，仮想アドレス空間じゃなくて物理アドレス空間，かな

mmap の例，確保領域のサイズは開始位置がずれる，のか．実行した結果は，たとえば，

```
7fb031a90000-7fb033e01000 rw-p 00000000 00:00 0
->
7faff1a90000-7fb033e01000 rw-p 00000000 00:00 0
```

こんな感じで，計算すると

```
(/ (- (- #x7fb031a90000 #x7fb033e01000) (- #x7faff1a90000 #x7fb033e01000)) 1024.0 1024.0) => 1024.0
```

たしかに 1GiB 増えてる

デバイスアクセス

p.135 あたりの loop device 使うあたり，開放はしなくていいのかな？？

```
fallocate -l 1G loopdevice.img
sudo losetup -f loopdevice.img
losetup -l
...
losetup -d /dev/loop0 # loop0 だったとして
losetup -l
rm loopdevice.img
```