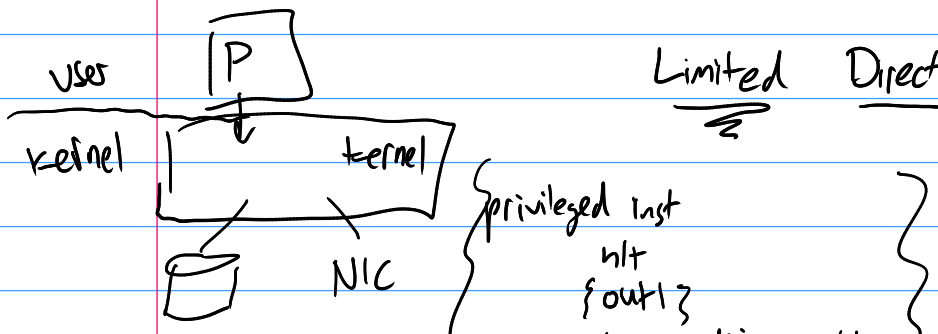
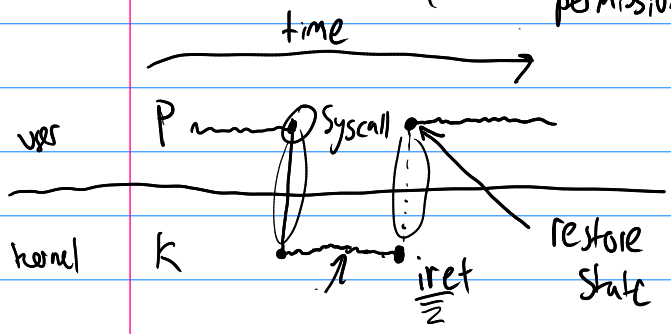
CS 3214 Lecture #3 more processes & unicode

Process : abstraction of a running program

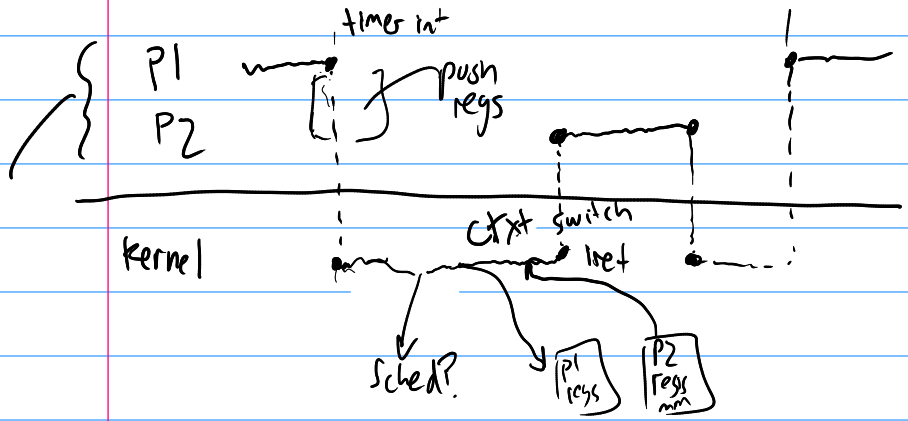




address space

registers
PC
SP

multiprogramming / time sharing



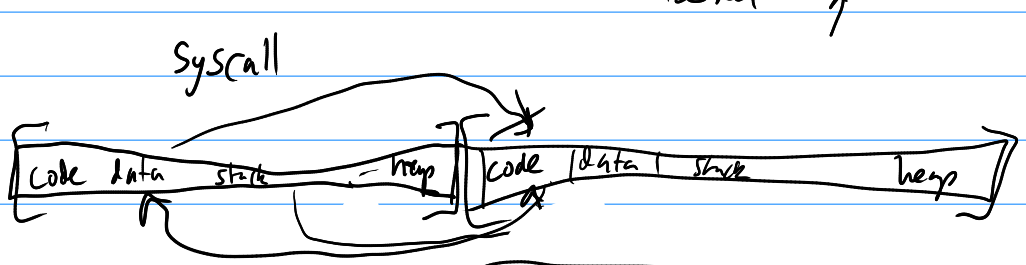
which P goes next:
kernel chooses
(scheduler)
not all Processes are ready

Process States

[address space change]
{TLB

vdso
vsyscalls
io_uring
BPF

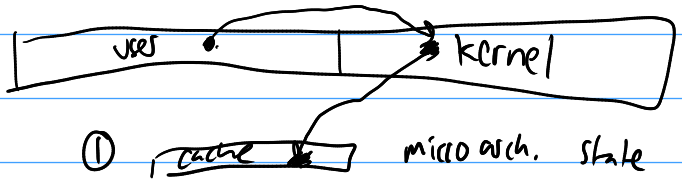
ctxt switch
- expensive



context switch vs. mode switch

code data stack heap

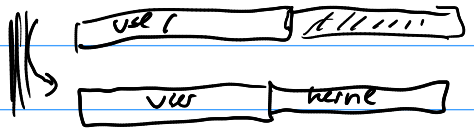
2018 Meltdown side channel / transient execution



① cache micro arch. state

② no permission, causes fault

{KPTI kernel page table isolation



Process States

[Unicode] what is character & how do you represent it?

old days ASCII 7-bits

1 char = 1 byte

0x41 = A

Unicode: character = Code points $2^{21} \simeq 2,\ldots$

1, 114, 112 of them

2^{32} 32 bit integers

UTF-32 : 32 bits for each char → python 3, Linux wchar_t, C11 char32_t

---→ really space inefficient 4x more space than ascii

+ all code points

+ uniform indexing over a string S[i]

UTF-16 → 2 bytes for most, 4 bytes for some

Java, Javascript → charSequence. Code Points

+ saved some space

− s[i]

error-prone

UTF-8

- variable length encoding

+ space efficient if are using common code points

(ASCII strings are valid UTF-8)

- indexing is hard ~~s[i]~~

given bytes, can you tell what encoding? NO!

- can tell if it's invalid

- use metadata Content-type: text/html; charset = UTF-8