

# d-kami OS について

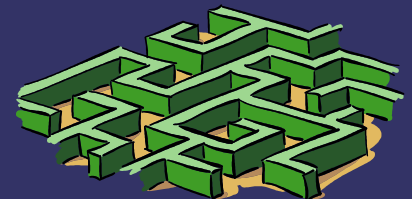
8/22

d-kami



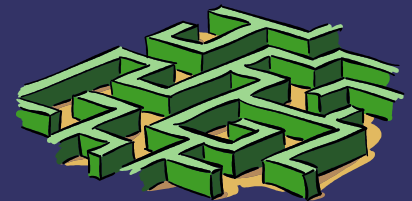
# 自己紹介

- ⇒名前： 上川大介
- ⇒所属： 筑波大学某研究室
- ⇒はてな： d-kami
- ⇒Twitter: d\_kami
- ⇒普通の大学院生



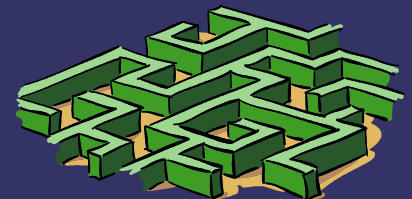
# 普通じゃない大学院生の例

- ⇒ Vim=VM
- ⇒ パイルダーオン
- ⇒ @syonbori



# OS を作ろうと思ったきっかけ

- ⇒ 2ch の『OS をつくりよう』スレを見て作ろうと思った
  - 自分にはできないのではないかな？
    - 失敗するものだと思ってやってみるか



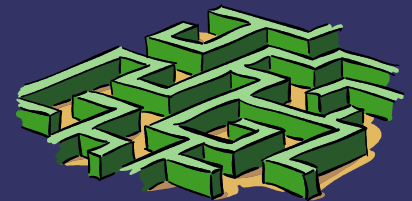
# 持ってた知識・技術

⇒ C 言語を使ったことがある（学校で習った程度）



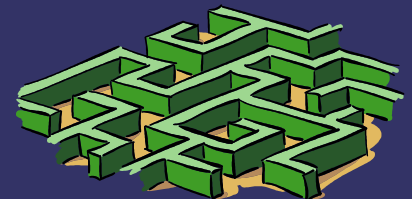
# 持ってなかった知識・技術

- ⇒ アセンブリ言語
- ⇒ x86 って何？
- ⇒ CPU のレジスタとか当然知らないし
- ⇒ 特権レベルとかなんのこっちゃ
- ⇒ ブートローダ（名前だけは知っていた）



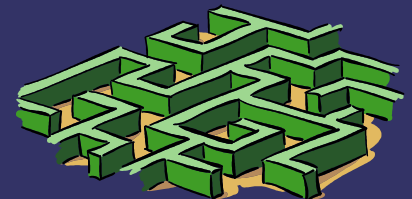
# 挫折経験

- ⇒ 『はじめて読む 486 』を読もうとして挫折
- ⇒ MINIX 本読もうとして挫折



# とりあえず作業開始

- ⇒ アセンブリ言語として NASM 利用
- ⇒ 最初は遊んでみる
- ⇒ INT 0x10 と INT 0x13 あたりを覚える





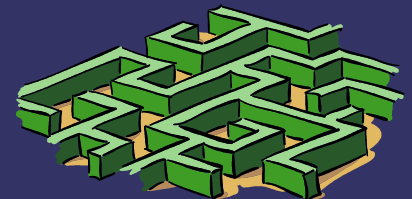
# INT 0x10 と 0x13

## ⇒INT 0x10

- グラフィック関係の機能が集まっている
- AH に 0x0e 突っ込んで AL に文字コード入れる
- BH に 0 入れて BL にカラーコード入れて INT 0x10
  - 文字が表示される

## ⇒INT 0x13

- AH に 2 を入れて、いろいろ設定すれば、ディスク読み込みができる



# A20 ゲートの有効化

- ⇒ アドレスバスの 21 本目のピン
- ⇒ A20 ゲートが無効化されている
  - メモリの 1MB 目が無効化されている
- ⇒ こいつ有効化しましょう
- ⇒ OUT 命令使ってればそのうち有効化できるでしょう



# 32bit モードへの道

## ⇒GDT の設定

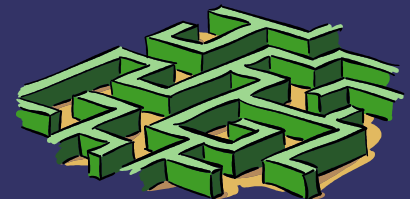
- セグメントディスクリプタにセグメントのベースアドレスやリミットを設定する
- メモリのアクセス権を設定する
- 32bit モード用のビットがあるので立てる

## ⇒LGDT 命令で GDT の場所とサイズを CPU に教える

## ⇒CR0 レジスタの 1bit 目を 1 にする

## ⇒セグメントレジスタにディスクリプタを指すように設定

## ⇒カーネルへ far JMP



# 32bit モード後の文字列表示

## ⇒32bit モードでのテキスト表示

- グラフィックモードによるけど TextVRAM 使えばできるよ

