

へにゃぺんて活動報告 2024 年春号

[著] 大神祐真

第十回 技術書同人誌博覧会 新刊
2024 年 5 月 12 日 ver 1.0

■免責

本書は情報の提供のみを目的としています。

本書の内容を実行・適用・運用したことで何が起きようとも、それは実行・適用・運用した人自身の責任であり、著者や関係者はいかなる責任も負いません。

■商標

本書に登場するシステム名や製品名は、関係各社の商標または登録商標です。

また本書では、™、®、©などのマークは省略しています。

はじめに

本書をお手に取っていただき、ありがとうございます！最近、「DaisyOS」という自作 OS のゲームボーイ版の実装を進めています。それ自体はまだ完成とは呼べないものですが、できることはありますので、これを作るに至った近年のへにゃぺんての「自作 OS」に対する考え方と共に「活動報告」という形でまとめてみました。

独自の理論に基づく独自の OS を紹介する内容となっていますので、「こんな考え方もあるのか」と面白がっていただけたら幸いです。

本書の更新情報等について

本書を含め、当サークルの同人誌・同人作品の情報は下記の筆者ウェブページにまとめています。

- <http://yuma.ohgami.jp/>

本書の内容について訂正や更新があった場合もこちらのページに記載します。何かおかしな点があった場合等は、まずこちらのページをご覧ください。

目次

はじめに	i
第 1 章 へにゃぺんての「自作 OS」について	1
1.1 「独自の OS」とは	1
1.2 DaisyOS とバイナリ生物学	2
第 2 章 DaisyOS GB v0.3	3
2.1 公開場所と実行方法	3
2.2 「ガイア理論」と「デイズワールド」について	6
2.3 生物的な振る舞いで自己制御している仕組み	8
2.4 操作例：デイズを間引く	11
2.5 操作例：地表温度に外乱を与えてみる	12
2.6 操作例：各デイズのメタデータ等を見してみる	13
2.7 操作例：評価関数を変えてみる	15
[コラム]白/黒デイズの命令列について	17
おわりに	19

第 1 章

へにゃぺんての「自作 OS」について

この章ではへにゃぺんての「自作 OS」について重視している考え方を説明します。この考え方が次章で紹介する「DaisyOS」へ繋がります。

1.1 「独自の OS」とは

へにゃぺんては「自作 OS」を軸に興味で活動している個人サークルです。「独自の OS」であること、特に「自分の OS」と呼べるようなものになることを重視しています。

2017 年に初の同人誌を出した「OS5^{*1}」という自作の OS の頃からその事を重視してはいました。「OS5」はブートローダー・カーネル・ユーザーランド (アプリケーション群) を全てフルスクラッチした OS です。簡単なものではありませんがフルスクラッチしていることで「自分の OS」と呼んでも良いかな、という思いでした。これは自身の勉強のために作ったものでもあり、OS の設計は現代の一般的な OS と同様に、メモリ管理・タスク管理・デバイスドライバやファイルシステムを有するものです。そして、ユーザーランドへ「プロセス」や「ファイル」といった抽象的な概念をシステムコールを通じて提供する、というものでした。

ただ、結局の所、中身の設計が一般的な OS と同じようなものなのであれば、それは一般的な OS の亜種に過ぎないのではないかと感じてしまいます。そして、「これ

^{*1} <http://funlinux.org/os5/>

こそが自分の OS だ」と呼べるような独自性の高い OS を作るにはどうすれば良いだろうかと考えるようになりました。

現代の大抵の OS は共通して『ファイル』や『プロセス (アプリ)』といった抽象的な概念をユーザーに提供する」というものであると思います。「ファイル」や「プロセス」は基本的にハードウェア側には無い概念です。単なる 0 と 1 のバイナリ列を「これは画像ファイル」、「これはテキストファイル」、「これはアプリ」といったようにユーザーに対してあたかもそのような実体があるように OS が見せかけているだけです。ただ、「ファイル」・「プロセス (アプリ)」という抽象化はとても便利な概念であるが故に、大抵の OS は「ファイル」や「プロセス (アプリ)」を扱うものであるのだと思います。これは、OS がコンピュータの上に「ファイル」や「プロセス (アプリ)」という概念に基づく「世界観」を提供しているとも言えると思います。

ただ、そもそもコンピュータは基本的に「メモリから命令を読んで CPU で実行する」という機械であり、『ファイル』や『プロセス (アプリ)』という世界観を提供する専用の機械」という訳では無いはずですが。どのような世界観を提供するかは OS 次第であり、「ファイル」でも「プロセス (アプリ)」でもない独自の世界観を提供する OS があっても良いのではないかと考えています。そして、そのようなものができれば、それこそが「自分の OS だ」と呼べるような独自性の高い OS になるのではないかと考えています。

1.2 DaisyOS とバイナリ生物学

「ファイル」でも「プロセス (アプリ)」でもない独自の概念として考えているのが「生物」です。「実行プログラムなどのバイナリ列を生物として扱う環境を提供する OS」というものを目指しており、それを実現する OS を「DaisyOS」と呼んでいます。

そして、それを実現するために、バイナリを生物として扱う仕組みを考え、設計や実装などを行う取り組みを「バイナリ生物学」と呼んでいます。具体的にどの様にして実行プログラムなどのバイナリ列を生物として扱うかについては、次章で「DaisyOS」の仕組みを説明する中で解説します。

第 2 章

DaisyOS GB v0.3

バイナリが生物のように振る舞う事のできるものの一つの例として、「ガイア理論」の「デイズワールド」をゲームボーイ (GB) 版の DaisyOS 上に実装してみました。この章ではそれを行った最近の製作物である「DaisyOS GB v0.3」を紹介します。

2.1 公開場所と実行方法

♣ 公開場所

「DaisyOS GB v0.3」は「itch.io^{*1}」というインディーゲームを公開するサイトで公開中です。

- URL : <https://yohgami.itch.io/daisyworld-demo-daisy-os-v030>
 - 「daisyos gb v0.3」等で Google 検索するか、あるいは筆者のウェブサイト^{*2}からもたどり着けます

♣ 実行方法

上記のページへアクセスすると図 2.1 の様に、「Run game」というボタンが表示されます。

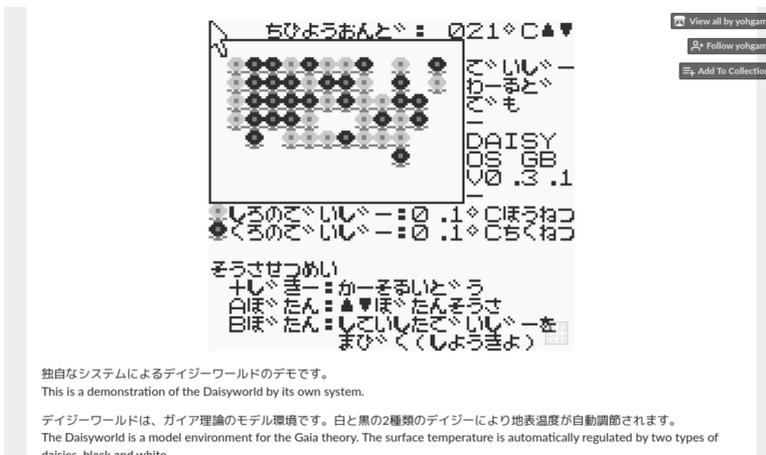
^{*1} <https://itch.io/>

^{*2} <http://yuma.ohgami.jp/>



▲ 図 2.1: 「DaisyOS GB v0.3」のページを開いた直後

このボタンを押すと図 2.2 の様に実行を開始できます。



▲ 図 2.2: 実行の様子

❁ 遊び方

基本的に見ているだけのソフトです。後ほど簡単に説明しますが、「ガイア理論」の「デイズワールド」が動いており、四角の枠内の白と黒の花（デイズ）の増減によって画面右上の「ちひようおんど（地表温度）」が、上下しつつもデイズの生育適

温 (20 °Cとしています) 付近に自己制御されます。

ROM ファイルも公開中

ページを下へスクロールすると、図 2.3 の箇所から ROM ファイルのダウンロードも可能ですので、お手元のエミュレータで試すこともできます。



▲ 図 2.3: ROM ファイルの公開場所

スマートフォンでも動きます

ブラウザで動作する「binjgb^{*3}」というエミュレータを使用しています。このエミュレータはスマートフォンにも対応しており、筆者が確認する限り Android の Chrome ブラウザでも動作します。

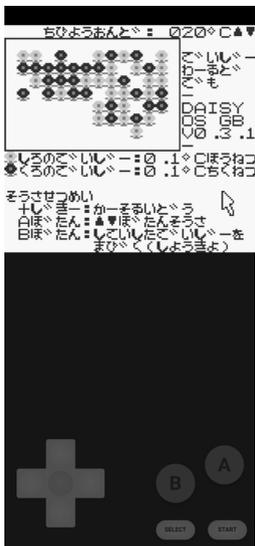
スマートフォンで試す際は、「Run game」ボタンをタップした後、GB 画面の右下に表示される最大化のボタン (図 2.4) をタップして実行画面を最大化してください。



▲ 図 2.4: 最大化ボタン

すると、最大化した画面の下部にタップで操作するコントローラが表示されます (図 2.5)。最大化しなくてもスマートフォンのブラウザで開けばタップで操作するコントローラは表示されているのですが、GB の画面表示自体に重なってしまっているため最大化した方が操作しやすいです。

^{*3} <https://github.com/binji/binjgb>



▲ 図 2.5: スマートフォンのブラウザ上で最大化した状態

.....

2.2 「ガイア理論」と「デージーワールド」について

このソフトは「ガイア理論」の「デージーワールド」というものを、一部独自の解釈も含みますが、GB版のDaisyOS上に実装したものです。「ガイア理論」と「デージーワールド」が何なのか分からないとこのソフトが何をやっているのかも分からないため、それぞれについて簡単に説明します。

.....

「よりシンプルなバイナリ生物学の本」を既読であれば次節の「PDP-7版との違い」の項まで飛ばして構いません

以降から次節の「PDP-7版との違い」の項までは、文章自体は書き直していますが、内容としては既刊「よりシンプルなバイナリ生物学の本」と同じです。そのため、その既刊を既に読んでいただいているようであれば、次節の「PDP-7版との違い」の項まで飛ばして構いません。

.....

♣ ガイア理論とは

「ガイア理論」は、「ガイア仮説」とも呼ばれるもので「生物は地球と相互に関係し合い、自身の生存に適した環境を維持するための自己制御システムを作り上げている*4」とする仮説です。

♣ デージーワールドとは

「デージーワールド」は、ガイア理論の妥当性を示すために考えられたモデル環境です*5。そこでは、まず生物としては白と黒の2種のデージーしか居ない世界を想定します。加えて、これらの2種のデージーの生育適温は同じであるとします。すると、この世界に降り注ぐ太陽光の量が増えた時、白いデージーは白が光を反射することで放熱の効果があるため生き続けることができますが、逆に黒いデージーは自身の黒が蓄熱の効果があるために生育適温を超えて暑くなりやすく死滅していきます。したがって、太陽光の入射量が増え、デージーが居ない地面では単純に地表温度が上昇していく状況でも、デージーが存在することで地表温度がデージーの生育適温付近で一定に保たれると考えられます。逆に、太陽光の入射量が減った場合も、黒いデージーが蓄熱の効果により生き続けることができ、逆に白いデージーが死滅していく事から、同様に地表温度が保たれます。これらは正にデージー自身の生存に適した環境を維持するための自己制御システムができていると考えられ、ガイア理論の妥当性を示しているとされています。

♣ 一部独自解釈している部分

デージーワールドを DaisyOS 上で表現するにあたり、実装の都合で以下の独自解釈をしています。

- 「太陽光」という要素は登場しない
 - 白いデージーは地表温度を下げ、黒いデージーは上げるというだけ
- 生育適温から外れても死ぬことはない
 - 死は寿命のみ
 - 生育適温から外れると増殖しにくくなる
 - * 暑くなるに連れ、黒いデージーは増殖しにくくなる
 - * 逆に寒くなるに連れ、白いデージーは増殖しにくくなる

*4 [ref.https://ja.wikipedia.org/wiki/ガイア理論](https://ja.wikipedia.org/wiki/ガイア理論)

*5 [ref.https://ja.wikipedia.org/wiki/デージーワールド](https://ja.wikipedia.org/wiki/デージーワールド)

2.3 生物的な振る舞いで自己制御している仕組み

❁ 概要

前章で説明した「実行プログラムなどのバイナリ列を生物のように扱う」事をどのように表現しているかという、まず、実行画面の枠線内の白と黒の花 (以降、それぞれを「白デイジー」・「黒デイジー」と呼ぶことにします) がそれぞれ小さい実行プログラムになっています。白デイジーは、白は光を反射する事から放熱の効果があるということで、地表温度を少し (0.1 °C) 下げます。逆に黒デイジーは、黒は光を吸収する事から蓄熱の効果があるということで、地表温度を少し (0.1 °C) 上げます。「地表温度」は内部的には単なる変数です。以上より、まとめると、白デイジーは地表温度という変数から 0.1 を減算するプログラムで、黒デイジーは逆に 0.1 を加算するプログラムです。^{*6}

そして、「生物のように扱う」といっても想定するのは単細胞生物程度の振る舞いです。具体的には細胞分裂と突然変異です。単細胞生物は自身のコピーを作ることで増殖します (細胞分裂)。そして、時々、その際にどこかがバグります (突然変異)。今回の白と黒のデイジーとしては、大抵の場合、白デイジーは白デイジーを産み、黒デイジーは黒デイジーを産みますが、時々、白デイジーが黒デイジーを、黒デイジーが白デイジーを産む事もある、という訳です。

このようにして、白と黒のデイジーが勝手に増えていく訳ですが、他に「実行と評価」と「死」の仕組みがあります。「実行と評価」は、周期的に環境上の全ての実行プログラムを実行し、それぞれを評価関数で評価する仕組みです。この評価関数から得られる値を「適応度」と呼んでおり、適応度が高いほど増殖しやすくなるようにしています。今回の場合、白/黒のデイジーにより地表温度を 0.1 °C 下げた/上げた結果が生育適温 (20 °C としています) より高ければ白デイジーの適応度を高くし、逆に低ければ黒デイジーの適応度を高くするように評価関数を設定しています。これにより、生育適温より暑い状況では放熱の効果がある白デイジーの方が増殖しやすく、逆に寒い場合は蓄熱の効果がある黒デイジーが増殖しやすくなります。

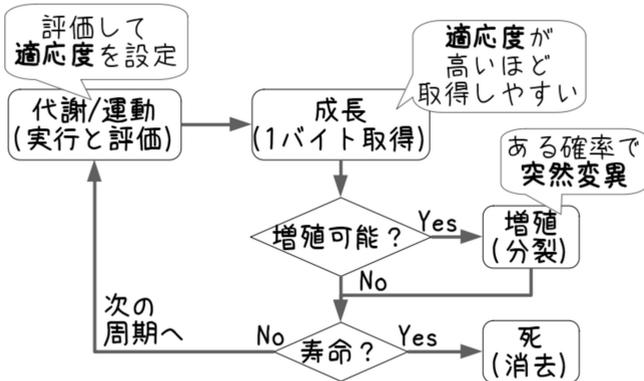
また、「死」の仕組みにより、環境上の全ての実行プログラムは無限に存在し続ける事はなく、いずれ消去されるようになっています。

以上の仕組みにより、地表温度を生育適温付近へ自己制御することが可能になります。

^{*6} 実際は直接「地表温度」の変数へ 0.1 の加算/減算を行うのではなく、別の変数へ 1 ずつ加算/減算を行い、10 あるいは -10 に至ったら「地表温度」変数へ 1 を加算/減算する、という事を行っています。ただ、本質的には「地表温度」変数へ 0.1 を加算/減算している事と変わらないので、説明の簡単のため「地表温度変数への 0.1 の加算/減算」として説明します。

♣ 各デイズの周期動作

前項の内容を含めた各デイズ（実行プログラム）の周期動作をまとめると図 2.6 の通りです。



▲ 図 2.6: 実行プログラムの周期動作

「代謝/運動」・「成長」・「増殖」・「死」の振る舞いを実行プログラムが行うようにしています。

「代謝/運動」は具体的には「実行と評価」です。前項で説明した通り、白/黒のデイズを実行し地表温度へ 0.1℃の減算/加算を行います。そしてその結果の地表温度に応じてそれぞれのデイズへ適応度を設定します。

適応度は次の「成長」のフェーズで環境から「1バイト」を取得できるか否かに関わります。この「1バイト」は何かというと、実行プログラム自身の命令列のいずれかの1バイトです。GB版 DaisyOSでは、それぞれの実行プログラムが自身を構成する命令列をもう一つ揃えると「増殖可能」となります。即ち、適応度が高いほど「成長」のフェーズでどんどん「1バイト」を取得できるため自身を構成する命令列を揃えやすく、増殖しやすいという訳です。

そして、増殖の際、ある確率で突然変異します。前項で説明した通り、突然変異は白デイズが黒デイズを産むかその逆のいずれかです。なお、突然変異確率は一定ではなく、適応度が低いほど突然変異しやすくなるようにしています。これは、環境の変化に対して生物が突然変異による進化の仕組みで適応していく様子を表現しているつものものです。元々環境に適応していればそれほど突然変異は起こらないと思われませんが、「突然地球全体が極寒に覆われる」など、環境が変化し生きにくい状況になると、恐らく細胞分裂時のエラーも起きやすく、突然変異が起きやすくなるので

はないかという考えです。

最後に「寿命を迎えたか」を確認します。寿命はこの周期動作を何回繰り返せるかというサイクル数で設定されます。現状は適応度などに関わらず全て同じで30サイクルとしています。そして、寿命を迎えると「死」としてそのデイジー(実行プログラム)は消去されます。

GB版DaisyOSでは以上の周期動作を実行プログラムに対して実施することで、実行プログラムを生物のように扱います。そして、それによりデイジーワールドによる地表温度の自己制御ができ、ひいては、実行プログラムを生物のように扱うことでこういった制御を行う事もできるという一つの例になっています。

♣ PDP-7版との違い

PDP-7版との設計の違いとして、実行プログラムが持つメタデータの違いがあります。PDP-7版では設計の簡単化のために「エネルギー」というGB版には存在しない項目がメタデータ内にあります。これはその名の通りでもありますが、「これが0になると死ぬ」というものです。PDP-7版ではこれを用いて、「成長」のフェーズでは適応度に比例したエネルギーを獲得できることとしています。そして、増殖のコストをエネルギーから払って増殖することとしました。^{*7}

「エネルギー」という項目によってどの様に簡単化したかと言うと、GB版の実行プログラムにメタデータとして存在する「取得フラグ」・「寿命」・「余命」という項目が「エネルギー」という項目により不要になります。

まず、「取得フラグ」は、GB版において「成長」のフェーズで取得済みのバイトを示すビットフィールドです。GB版では「自身を構成する命令列をもう一つ揃えると増殖可能」であるためこの様なビットフィールドが必要になります。しかし、PDP-7版は「増殖のコスト以上のエネルギーがあれば増殖可能」であるため「取得フラグ」は不要になります。

また、「寿命」・「余命」はそれぞれ「実行プログラムが生まれてから死ぬまでのサイクル数」と「現在の残りサイクル数」です。項目を分けている理由は後述します。PDP-7版では「エネルギーが0になると死ぬ」としている事に加え、1サイクル毎に「成長」で獲得するエネルギー量より多いエネルギー量を「定常的なエネルギー消費」として消費するようにすることで、実行プログラムが無限に生き続けることが無いようにしています。これにより「寿命」・「余命」という項目も不要になります。

主な違いとしては以上の通りです。ただ、これらはあくまでも簡単化のためのものであり、本質的な設計の違いはありません。

^{*7} 増殖のコストは比較的高めにしています。

2.4 操作例：デイジーを間引く

基本的に見ているだけのソフトですが、ユーザーが行える操作も少しあります。以降の節ではそれらを紹介します。

まず行える操作として、画面上のデイジーを間引く(消去)事ができます。十字キーでマウスカーソルを動かして枠内のいずれかのデイジーに合わせてBボタン(PCのブラウザ上の場合はZキー)を押してみてください。すると、マウスカーソルで選択されたデイジーが間引かれて画面から消えます。

これにより、「意図的に白デイジーを減らして黒デイジーばかりにしてみる」(あるいはその逆)といった実験を行ってみることができます。試しに白デイジーをどんどん間引いて黒デイジーばかりにしてみると、黒デイジーの蓄熱が白デイジーの放熱に勝ることで地表温度を上昇させることができます(図 2.7)。



▲ 図 2.7: 白デイジーを間引いて地表温度上昇

ただし、だんだんと高温では黒デイジーが増殖しにくく、逆に白デイジーが増殖しやすくなることから、ユーザー操作で白デイジーを間引くより早く*8白デイジーが増殖していくようになり、急増した白デイジーにより地表温度がまた生育適温付近へ戻されます(図 2.8)。

*8 頑張れば白デイジーの増殖より早く間引いていくこともできるかもしれませんが。



▲ 図 2.8: 反動による白デイジー増殖で生育適温付近へ戻される

ただ、これはピンと張った弦を引っ張って弾いたようなもので、急増した白デイジーによる温度低下は生育適温を超えた温度低下を招きます。今度は低温で白デイジーが増殖しにくく、黒デイジーが増殖しやすくなることで地表温度が上昇し、という形で何度か揺戻しがありながらもだんだんと生育適温付近を保つようになっていきます。

2.5 操作例：地表温度に外乱を与えてみる

また実は、地表温度はユーザー操作で上げたり下げたりといった外乱を与えることができます。地表温度の操作は画面右上の「ちひようおんど」の欄の右側にある「▲」と「▼」のボタンで行えます(図 2.9)。「▼」ボタンにマウスカーソルを合わせて A ボタン(PC 上の場合は X キー)で 1 °C 上げ、「▼」ボタンで 1 °C 下げます。ボタンのリリースのタイミングで処理を行っており、押しっぱなしには対応していません。^{*9}



▲ 図 2.9: 地表温度を操作する▲・▼ボタン

(ちょっと大変ですが、)「▼」を連打して-100 °Cくらいまで強制的に下げると、

^{*9} なので、いっぱい温度を上昇/下降させたい場合は頑張って連打していただく必要があります。

前節とは逆に白ダイジーが減り、そして黒ダイジーが増えて地表温度を上げる方向へ自己制御が働く様子を確認できます (図 2.10)。



▲ 図 2.10: 地表温度を -100 °C 辺りまで下げ、黒ダイジーばかりになっている様子

2.6 操作例：各ダイジーのメタデータ等を見ている

♣ 開発版について

この節と次節の内容は 2024 年 3 月時点では開発版で公開している内容です。開発版は以下で公開しています。

- URL : <https://yohgami.itch.io/daisyos-dev>
 - 「daisyos dev」等で Google 検索するか、あるいは筆者のウェブサイトからもたどり着けます

「Run game」を押して起動した直後の見た目に違いはありませんが、セレクトボタン (PC で実行中の場合、TAB キー) でダイジーを表示している枠外の表示を以下の順に切り替えることができます。

1. ソフト説明表示 (デフォルト)
2. 実行プログラムの情報表示
3. 評価関数設定表示
4. 1. へ戻る

なお、スタートボタン (PC で実行中の場合、Enter キー) で「ソフト説明表示」へ

戻すことができます。

❁ 実行プログラムの情報表示について

ここでは、「実行プログラムの情報表示」を紹介します。画面表示をこの表示にした状態で、マウスカーソルをいずれかのデ이지ーに合わせ、Bボタン(PCで実行中の場合、Zキー)を押してみてください。すると、マウスカーソルで指定したデ이지ーの情報が表示されます(図 2.11)。



▲ 図 2.11: 実行プログラムの情報表示例

表示されているそれぞれの項目について説明します。まず「ふらぐ(フラグ)」は、その名の通り各種のフラグが設定されるビットフィールドです。ただ、現状は最下位ビットがこの実行プログラムの生死を示しているのみ(1の場合、生きています)であり、表示されている実行プログラムは全て生きていますので常に1です。

「さゝひよう(座標)」は、この実行プログラムがデ이지ーとして存在している座標です。座標値は、画面左上を原点とした8x8ピクセルのタイル単位の値です。

「よめい/じゆみよう(余命/寿命)」もその名の通りですが、「余命」は現在から死までのサイクル数で、「寿命」は生まれてから死までのサイクル数です。前述の通り「寿命」は全ての実行プログラムで共通で30サイクルです。そのためこのデ이지ーワールドの実装の場合、「寿命」をそれぞれの実行プログラムで持っている必要は無いのですが、過去の実装では突然変異の際に寿命も変化させていた事がありました。その際は親の寿命をベースとしており、突然変異しなかった場合も親の寿命を子の寿命としていたため、毎サイクルでデクリメントされる余命とは別に寿命をそれぞれの実行プログラムで保持しておく必要がありました。

「てきおうど (適応度)」は、前述の通り、評価関数で評価した結果の評価値です。

「ばいなり (バイナリ)」と「さいず (サイズ)」は、実行プログラムの命令列自体とそのバイト数です。サイズは白デージー・黒デージー共に5バイトで、たったそれだけのサイズですが、白デージーでは「地表温度を下げる」、黒デージーでは「地表温度を上げる」命令列になっています。それぞれの命令列について詳しくは章末のコラムをご覧ください。

「しゅとくふらぐ (取得フラグ)」は、自身の命令列の各バイトを取得したことを示すビットフィールドです。白/黒デージー共に5バイトの命令列なので、0X1F(16進数で下位5ビットが1になった状態)になると増殖可能となります。

2.7 操作例：評価関数を変えてみる

♣ 評価関数を変えてみる

それでは次に「評価関数設定表示」について説明します。この表示に切り替えた状態は図 2.12 の通りです。



▲ 図 2.12: 評価関数設定表示状態

ここでは評価関数を別のものへ変更して試してみることができます。画面下部の「ひようかかんすうせんたく (評価関数選択)」の枠内にあるものが選択できる評価関数です。この表示ではマウスカーソルが消え、上下キーによりこの評価関数を切り替えることができます。デフォルトでは「でいじーわーど (デージーワールド)」が選択されていますが、下キーで「つねに 255(常に 255)」へ変更してみてください。

♣ 各実行プログラムの適応度がどうなったか見てみる

「つねに 255(常に 255)」は、その名の通り、常に適応度として最大の 255 を返す評価関数です。この関数へ変更したことで、各実行プログラムの適応度がどうなったか見てみましょう。再度、「実行プログラムの情報表示」へ切り替えてください。

そして、適当なデイジーを B ボタンで選択すると、どのデイジーも適応度が 0XFF(255) になっていることが確認できます (図 2.13)。



▲ 図 2.13: 適応度が 0XFF(255) になっている

♣ 地表温度変化の外乱を与えてみる

評価関数が「常に 255(最大) を返す」もの変わったというのはどういう状況かというと、イメージとしては「白は放熱し、黒は蓄熱するというルールが無くなり、いずれのデイジーも温度によらず常に最適に生存している」という状況です。実行プログラムの周期動作で言うと、「成長」のフェーズで必ず 1 バイトを取得できる事になり、白/黒いずれのデイジーも増殖しやすい状況になります。^{*10}

それでは、「2.5 操作例：地表温度に外乱を与えてみる」で試したように画面右上の「▼」にマウスカーソルを合わせて A ボタン (PC の場合 X キー) を連打し、地表温度を -100 °C 程まで下げてみてください。すると今度は以前のように黒デイジーばかりになるような事も無く、地表温度の自己制御も働きません (図 2.14)。^{*11}

^{*10} ただし、メモリサイズと実装の都合上、存在できる実行プログラムの数は 50 個が上限です。そのため、枠内の全てのマスがデイジーで埋まることはありません。

^{*11} ただし、たまたま黒デイジーが多い状況だったりすると、地表温度が上がっていくことはあるかと思えます。



▲ 図 2.14: 今度は地表温度を-100°C付近まで下げても黒デージーばかりになったりしない

この様に、評価関数を切り替えることで、個々の実行プログラムにより全体としてどのような振る舞いをさせるのかを変えることができます。

【コラム】白/黒デージーの命令列について

白/黒デージーの命令列を解説します。まず白デージーの命令列はアセンブリ表記を併記すると表 2.1 の通りです。

▼ 表 2.1: 白デージーの命令列

CPU 命令	アセンブリ表記
0x21 34 c0	LD HL, 0xc034
0x35	DEC (HL)
0x00	NOP

3つのCPU命令です。まず1つ目のLDという命令ではHLという16ビットのレジスタ*12へ0xc034という値を設定しています。なお、「レジスタ」とはメモリ(RAM)よりも高速ですが小容量な記憶領域です。CPU命令のレベルでは主にレジスタを対象に演算を行います。(必要に応じてメモリへもアクセスします。)

2つ目のDEC命令では、レジスタHLに設定されている16ビットの値をアドレスとし、そのアドレスが指す先の値をデクリメントしています。なお、レジ

スタ **HL** に設定していた「0xc034」は地表温度の変数を1ずつ増減させる前段の変数のアドレスです。(前述もしましたが、) この変数をインクリメント/デクリメントしていき、10あるいは-10に達したら地表温度の変数を1ずつ増加/減少させています。これにより、デイズ1つあたりの影響が0.1℃となるようにしています。

3つ目の **NOP** 命令は「何もしない」命令です。実装上、各実行プログラムの命令列は5バイトだと都合が良いため、この「何もしない」命令を置いています。そして、黒デイズの命令列は表 2.2 の通りです。

▼表 2.2: 黒デイズの命令列

CPU 命令	アセンブリ表記
0x21 34 c0	LD HL, 0xc034
0x34	INC (HL)
0x00	NOP

2つ目が **INC** という命令になっている他は白デイズの命令列と同じです。**INC** 命令では、レジスタ **HL** に設定されているアドレス先の値をインクリメントしています。これにより、黒デイズでは地表温度を0.1℃上昇させます。

*12 正確には H という 8 ビットのレジスタと L という 8 ビットのレジスタの 2 つなのですが、これらを合わせて 1 つの 16 ビットのレジスタとして使うことができますようになっています。

おわりに

ここまで読んでいただきありがとうございます！本書では近年のへにゃべんての自作 OS についての考え方を紹介し、最近の製作物である GB 版 DaisyOS の現時点のバージョンでできることを紹介しました。まだできることは少ないですが、「実行プログラム等のバイナリを生物として扱う」事で何かを自己制御するような事が可能であることを示すことはできたかと思います。興味深いと考えているポイントは「何らかの制御を行っているプログラムが直接的に存在するわけではない」という所です。実行プログラムをちゃんと生物として扱う事で結果的に自己制御がなされていきます。今回で言うと、「変数から 0.1 を減算するプログラム」と「変数へ 0.1 を加算するプログラム」を、白と黒のデージーとして、「白は放熱」し「黒は蓄熱」という所も含めてちゃんと生物として扱うと、結果として地表温度が生育適温付近へ自己制御されていくようになります。

DaisyOS は何ができる OS なのかという事を明確に言葉で表現することは、まだ考え中な所もあり難しいですが、デージーワールドで確認したように、生物として扱うことで結果的に起こる「何か」が「DaisyOS でできる事」なのだと考えています。基本的には、環境上に存在するバイナリを環境設定(評価関数)に応じて進化させていく、というものなので、「とある変数の制御」以外にもできることは色々あるだろうと考えています。

ただし、現状の GB 版 DaisyOS ではユーザーができる操作としては、外乱を与えたり、評価関数を変更できたりするくらいなので、もっとユーザーのできることを増やしていきたいと考えています。例えば、現状では評価関数として 2 つの関数しか選べませんが、もっと工夫した関数を用意すれば、例えば「白/黒のデージーで枠内に何かを描く」といった事もできるのではないかと考えています。あるいはそもそもユーザーが評価関数を DaisyOS 上で自由に作れるようになればもっとできることが増えると思います。

他にも突然変異の際、現状では「0.1 を減算するプログラム」か「0.1 を加算するプログラム」にしか突然変異しませんが、「命令列上の任意の命令が別の命令に変わる」、あるいは「任意の命令が追加/削除される」、といった突然変異ができるようになれば、評価関数を満たすような任意のプログラムを進化の仕組みで生成することも可能です。^{*13}

DaisyOS は、筆者自身もまだ気づいていない未知の可能性を秘めた OS だと考え

^{*13} 実は過去にその様にして ELF 形式の実行バイナリを生成するツールを作成し実験していたことがありました。詳しくは既刊『バイナリ生成環境「daisy-tools」実験報告』をご覧ください。

ています。それを自分でも実際に見てみたいと、そして皆さんにお見せできるように、まず直近では GB 版でのユーザーの自由度を上げ、この OS 上で表現できることの幅を広げていきたいと考えています。

この様な特殊な (変な) 活動ではありますが、へにゃべんてを引き続き面白がっていただけると幸いです。

へにゃぺんて活動報告 2024 年春号

2024 年 5 月 12 日 ver 1.0 (第十回 技術書同人誌博覧会 新刊)

著 者 大神祐真

発行者 大神祐真

連絡先 yuma@ohgami.jp

<http://yuma.ohgami.jp>

@yohgami (<https://twitter.com/yohgami>)

印刷所 日光企画

© 2024 へにゃぺんて

(powered by Re:VIEW Starter)