

# 恐怖の迷宮への招待:

*NetHack* ガイドブック

オリジナル - Eric S. Raymond

(Mike Stephenson らによるバージョン 3.6 のための校訂と拡充)

2020年3月8日

## 1 はじめに

最近、あなたは日々の仕事に満たされなく、敬遠しつつある自分に気がついた。最近数ヶ月、あなたが見る夢は探査、盗み、聖戦、戦闘といったものであったが、あなたにはその理由がはっきりとはわからなかった。あなたはこれらの夢を実際には生まれてこのかたずっと見続けていて、なぜか今までそれらについて忘れようとしていたのではないか、と思った。ある夜、あなたは飛び起きて、夢の中の洞窟のあらゆる角の後ろに潜んでいるように見える、奇妙で強力な怪物の鮮明な記憶にぞっとして大声で叫んだ。あなたの夢に没入する内容は本当なのだろうか？夜毎に、廃墟の近くの不思議な洞窟に入るという欲求が強くなった。それでも、毎朝あなたはそのような考えを、洞窟に入って帰って来なかった人々の話を思い出すことで頭から追い出した。しかし、ついにあなたは夢の中に出てくる不思議な場所を捜し求めるといふ思いを抑えきれなくなった。何ととっても、冒険者は最初にその道を通っていった時より、こちらに戻って来る時の方が裕福になっているように見えるのだ。戻って来ない人たちは皆、単にまだ冒険を続けているだけなのでは？

あちこちで尋ねるうちに、あなたはある噂を耳にした。イェンダーの魔除けとかいうものがあり、手に入れることができたなら、すばらしい財産をもたらすと言うのだ。あなたが聞いたとある伝説では、魔除けを見つけたものは神から不死の体を授かるとさえ言われているらしい。魔除けは恐怖の迷宮の奥深く、ゲヘナの谷を越えたどこかにあるという噂である。伝説を聞いた瞬間あなたは、自分には洞窟に入って彼らの話している魔除けを捜し求めるべき深遠で未知のなんらかの理由があると確信した。あなたはたとえ魔除けの力が真実でなくても、少なくとも冒険談を地元の吟遊詩人にかなりの金額で売り込めるだろうと考えた。夢の中で出てきた恐ろしく、不思議な怪物に出会うことができばなおさらである。あなたは地元の安宿で最後の宿を取り対策を練った。安宿の壁に掲げられた成功率のオッズが下がれば下がるほど、あなたは意気消沈した。

朝起きると、あなたは所持品をかき集め、あなたは洞窟に向けて旅立った。無事何日かの旅を続けて恐怖の迷宮への入口を示す古代の遺跡へとたどり着いた。夜も更けていたので入口でキャンプをすることにし、その夜は広々とした空の下で眠りに就いた。翌朝あなたは道具をかき集め、もしかすると地上での最後になるかもしれない食事を済ませ、洞窟へと入っていった…

## 2 あなたを取り巻く状況

こうしてあなたは *NetHack* というゲームを始めることになった。あなたの目的は持てる限りの宝を集め、イェンダーの魔除けを見つけ出し、さらにはこの恐怖の迷宮から生きて脱出することである。

冒険におけるいろいろな困難に対応するための各種の能力や力は、あなたの生い立ちと修行によって変化する。

**考古学者 (Archeologist)** 洞窟について多くの知識を持っている。このため素早く動いたり洞窟での不快な事態を切り抜けることができる。彼らは学術的調査のための探検旅行に適した道具を持って出発する。

**野蛮人 (Barbarian)** 奥地からやってきた戦士であり、戦闘のために鍛えられている。彼らは類まれな力の強さを持ち、丈夫な鎖かたびらと大きな両手持ちの剣のみを持って探求に出発する。

**洞窟人 (Caveman と Cavewoman)** とりわけ強い力を持っているが、新石器時代に使われた武器を持って出発する。

**薬師 (Healer)** 医術や薬物について精通している。生気を回復させたり、苦痛を和らげたり、麻酔をかけたり、毒を中和させたりする香草や薬草について詳しい。そして器具を使って生物の

健康状態を見抜くことができる。彼らは開業医としてかなりよい報酬を得て、それを持って洞窟に入っていく。

- 騎士 (Knight)** 騎士道に忠実であることと防具が驚くほど優れていることという点で、普通の戦士とは区別される。
- モンク (Monk)** 修道者であり、厳格な肉体的習慣と精神的訓練によって、武器なしでも武器を使うのと同じように戦うことができる。モンクは防具をつけないが、機動性を向上させることで埋め合わせている。
- 僧侶と尼僧 (Priest と Priestess)** 聖職の戦士であり、武器と防具を持ち魔法の業を駆使して正義を広めんとする聖戦士である。祈りを通じて神と語る能力により、彼らはしばしば危難から逃れることができるが、祈りは危難をもたらすこともある。
- レンジャー (Ranger)** ほとんど森を住みかとし、洞窟は少々場違いかもしれない。しかし、彼らは追跡術や隠密行動と共に、弓術の達人である。
- 盗賊 (Rogue)** 敏捷で身を隠す業に長けた泥棒であり、錠や罨や毒に詳しい。彼らの長所は奇襲であり、これは非常な長所となる。
- 侍 (Samurai)** 封建時代の日本の精鋭の武人である。防具は軽装で身のこなしに秀でており、この上なく研ぎ澄まされた 大小 2本の刀を帯びている。
- 観光客 (Tourist)** 多額の金(買物に最適である)とクレジットカード、大量の食料、地図、そして高価なカメラを持って出発する。ほとんどの怪物は写真に撮られるのを嫌うものである。
- ワルキューレ (Valkyrie)** 勇敢な女戦士である。過酷なスカンジナビアの国々で育った彼女らは頑強で、厳しい寒さにも耐え、身を隠す業と巧妙さを身につけている。
- 魔法使い (Wizard)** 魔術の知識とに選抜かれた魔法の道具を準備しており、いにしへの魔法学に対して特に造詣が深い。見かけは弱そうで簡単に打ち倒せるように思えるが、経験を積んだ魔法使いは恐るべき敵である。

また、キャラクターの種族も選ぶことができる(制限はある;ほとんどの職業はどの種族がなれるかに制限がある):

- ドワーフ (Dwarf)** 人間やエルフよりも小さいが、がっしりした体つきで丈夫である。ドワーフの注目すべき特性は採掘と金属工作の専門的技術である。ドワーフの防具はエルフのミスリル防具と同じとまではいかないまでも、それに次ぐ品質である。
- エルフ (Elf)** 機敏さと迅速さ、そして鋭敏な感覚を持っている。周囲で起きるどんな些細なことも彼らの目を逃れることはできない。職人気質の彼らの持つ武器と防具は優秀なものである。
- ノーム (Gnome)** ドワーフよりもさらに小さいが、ドワーフと似ている。ノームは熟練した採掘師として知られ、富と危険で満ちている秘密の地下坑道を恐怖の迷宮に作っていることで知られる。
- 人間 (Human)** 地上で最も一般的な種族で、ゆえに他の種族と比較される時の基準となっている。特別な能力は持っていないが、あらゆる職業で成功することができる。
- オーク (Orc)** 残酷で野蛮な種族で、(他のオークを含む)あらゆる生物を嫌っている。特に、オークはエルフを感性の違いゆえに嫌い、わざわざあらゆる機会を見つけて殺そうとする。オー

クによって作られた防具と武器は一般的に品質が劣る。

### 3 画面に表示されるものの意味

画面上にはあなたが現在いる洞窟の階ですでに探索を終えた部分の地図と、そこで目にした物が示されている。さらに深く探検を進めるに連れて洞窟はその姿を次第にあなたの前の画面に明らかにしていく。

*NetHack* の祖先である *rogue* が最初に現れたとき、そのスクリーン指向はコンピュータファンタジーゲームの中でも全く独特のものであった。それ以来スクリーン指向は例外的なものというよりむしろ標準となった。*NetHack* はこのすばらしい伝統を受け継いでいる。疑似的な英語でコマンドを入力し、結果が文章で表示されるテキストアドベンチャーゲームとは異なり、*NetHack* のコマンドはすべて 1 文字か 2 文字のキー入力で与えられ、その結果は画面上にグラフィカルに表示される。画面は最低 80 桁および 24 行の大きさが推奨される。それ以上の大きさがあっても 21 × 80 の領域だけが地図の表示に使われる。

*NetHack* は点字や音声合成の助けを借りて、目の不自由な人でもプレイできる。目の不自由な人のための *NetHack* の設定方法については、このドキュメントで後述する。

*NetHack* はプレイするたびに新しい洞窟を作成する。このため何回かゲームに勝利したことのある作者たちにも、いまだこのゲームは楽しく興奮に満ちたものと感じられる。

*NetHack* はさまざまな表示のオプションを提供する。オプションはあなたのハードウェアやソフトウェアの能力に依存し、プラットフォーム毎にさまざまである。さまざまなコンパイル時のオプションは実行ファイルが作られたときのみ可能となる。モノクロキャラクタインタフェース、カラーキャラクタインタフェース、タイルと呼ばれる小さな絵を用いたグラフィカルインタフェースの 3 つの表示方法がある。2 つのキャラクタインタフェースはフォントを他の文字に置き換えることもできるが、標準設定では標準の ASCII キャラクタを用いて全てを表現する。さまざまな表示のオプションはゲーム内容には影響を与えない。タイルや色はガイドブックで表記できないし、それは全てのプラットフォームに共通ではないので、ゲーム中表示されるものの説明としてはモノクロキャラクタディスプレイで表示される標準の ASCII キャラクタを用いる。*NetHack* で何が起きているのかを理解するには、まず *NetHack* では画面がどうなっているのかを理解しなくてはならない。*NetHack* の画面は、テキストアドベンチャーゲームにおける「あなたは…を見た」のような文章の代わりである。図 1 は *NetHack* では画面に何が表示されるかの一例である。画面がどのように表示されるかはプラットホームに依存する。

```
The bat bites!
```

```
-----  
|...|   -----  
|.<..|####...@...$.|  
|...-#  |...B...+  
|...|   |.d.....|  
-----  -----|--
```

```
Player the Rambler    St:12 Dx:7 Co:18 In:11 Wi:9 Ch:15 Neutral  
Dlvl:1 $:0 HP:9(12) Pw:3(3) AC:10 Exp:1/19 T:257 Weak
```

図 1

## ステータス行 (画面の下部)

画面の下 2 行には、あなたの現在の状態を表す暗号のような情報の断片が含まれている。どちらかの行が画面の幅よりも長くなるとその全部を見ることはできないかもしれない。以下に状態を示すいろいろな項目が何を意味するのかを述べる (コンフィグレーションによっては、以下に一覧されている状態を示す項目のすべてが表示されるとは限らない)。

**ランク (Rank)** あなたのキャラクター名と (後述の経験レベルに基づく) 職業別の等級。

**強さ (Strength)** あなたのキャラクターの力の強さの値であり、6 つの基本的な属性の 1 つである。人間のキャラクターの場合、力の強さは 3 から 18 の間の値をとる。その他の場合、この制限を越えることもある (時折 18/xx のような特別な力を得ることもあり、魔法の力も普通の制限を越える能力を与えることがある)。力の強さの値が大きいほど、あなたの力は強い。力の強さは力仕事をどれだけうまくやれるか、戦いでどれだけダメージを与えるか、どれくらい荷物を運べるかに影響する。

**素早さ (Dexterity)** 素早さは戦いにおける命中率、罠を逃れる可能性、そしてその他の敏捷さを必要とする仕事や物をいかに器用に扱えるかなどのがらに影響を与える。

**耐久力 (Constitution)** 耐久力は怪我やその他の圧迫によるスタミナの消耗から回復する能力に影響する。強さが低いか中くらいの場合、どれだけものを運べるかに耐久力も影響する。十分に強さが高い場合、運搬能力に関して耐久力は関係なくなる。

**知力 (Intelligence)** 知力は呪文を唱える能力や魔法書を読む能力に影響を与える。

**賢さ (Wisdom)** 賢さは (特に魔法を扱う) 実戦的な経験により得られる。魔法のエネルギーに影響する。

**魅力 (Charisma)** 魅力はある種の生物があなたに対してとる態度に影響を与える。特に店主が示す価格に影響する。

**属性 (Alignment)** 属性には 秩序 (*Lawful*), 中立 (*Neutral*), 混沌 (*Chaotic*) がある。しばしば、秩序は善、混沌は悪であるとされるが、法と道徳が常に一致するとは限らない。あなたの属性は他の怪物のあなたに対する反応に影響を与える。属性が似ている怪物は好戦的ではなく、反対の属性に対してはあなたの存在に対して極めて好戦的であるらしい。

**階数 (Dungeon Level)** どれだけ洞窟の奥深くまで行ったかを示す。初期値は 1 で、洞窟を奥深く行くほど大きな値になる。いくつかの階は特別で、数字ではなく名前で識別される。イエンダーの魔除けは地下 20 階より深くのどこかにあると考えられている。

**金 (Gold)** 大抵に所持している金貨の数である。箱などに隠したお金は勘定に入っていない。

**体力 (Hit Points)** 体力の現在値と最大値である。体力はあとどれだけダメージを受けると死ぬかを示す。戦いで傷つくほど値は減少する。体力は休息や魔法のアイテムや呪文により回復できる。括弧の中の数字は体力のとり得る最大値である。

**魔力 (Power)** 呪文に使う魔力である。これは呪文を唱えるのに必要な神秘的力 (*mana*: マナ) がどれほどあるかを示す。魔力もまた休息することにより回復する。

**防御値 (Armor Class)** 非友好的な生物からの攻撃を防具がどれほど食い止められるかを示す値である。数字が小さいほど防具は有効であり、防御値が負の値になることさえ十分あり得る。

<b>経験値 (Experience)</b>	現在の経験レベルと経験値である。冒険が進むと経験値を得ることができる。経験値の合計がある一定値まで達すると、経験レベルが上がる。経験を積むにつれ、戦い方が上達し魔法の攻撃にも耐えられるようになる。多くのバージョンでは経験レベルだけが表示される。
<b>時間 (Time)</b>	経過したターン数である。time オプションがオンのときに表示される。
<b>状態 (Status)</b>	腹具合: 現在の腹具合状態。値は <i>Satiated</i> , <i>Not Hungry</i> (or <i>Normal</i> ), <i>Hungry</i> , <i>Weak</i> , <i>Fainting</i> 。 荷物: 運んでいる物があなたの移動能力にどれくらい影響を与えているかを示す。値は <i>Unencumbered</i> (通常)、 <i>Encumbered</i> (よろめき)、 <i>Stressed</i> (圧迫)、 <i>Strained</i> (限界)、 <i>Overtaxed</i> (過重)、 <i>Overloaded</i> (超過)。 <i>Unencumbered</i> のときには表示されない。 致命的な状態: <i>Stone</i> (石化)、 <i>Slime</i> (どろどろ; スライム化)、 <i>Strngl</i> (窒息)、 <i>FoodPois</i> (食毒)、 <i>TermIll</i> (病気)。 致命的でない状態: <i>Blind</i> (盲目)、 <i>Deaf</i> (耳聾)、 <i>Stun</i> (眩暈)、 <i>Conf</i> (混乱)、 <i>Hallu</i> (幻覚)。 移動状態変化: <i>Lev</i> (浮遊)、 <i>Fly</i> (飛行)、 <i>Ride</i> (騎乗)。 その他の状態も存在するが、それらの状態を表示するための空きがない。‘^X’ コマンドは関係する全ての状態を表示する。

## メッセージ行 (画面の上部)

画面の最上行は視覚的に表現できないことを説明するメッセージのために使われる。最上行に “--More--” が表示されたときは、このあとにさらにメッセージが続いているけれども現在のメッセージをまず確実に読むよう促しているということである。次のメッセージを読むためには単にスペースキーを押せばよい。

メッセージ行にどんなメッセージをどのように表示するかを変更するには、“*Configuring Message Types*” と *verbose* オプションを参照のこと。

## 地図 (画面のその他の部分)

画面のその他の部分は、あなたが現在いる階でそれまでに探索し終わった部分の地図である。画面上のシンボルはそれぞれ何かを表している。表示されるシンボルのうちいくつかは、さまざまなグラフィックオプションを設定することによって変えることができる。そうでなければ標準設定のシンボルが用いられる。次に標準設定のシンボルの意味の一覧を示す。

"- と	部屋の壁、または開いた扉、または墓 (I)。
.	部屋の床、氷、または扉のない出入口。
#	通路、鉄の棒、木、あるいは (あなたの迷宮にあるなら) 台所の流し台、または跳ね橋
>	上り階段: 上の階への通路。
<	下り階段: 下の階への通路。
+	閉じた扉、または学習できるかもしれない呪文の書かれた魔法書。
@	あなたのキャラクター、または人間。
\$	金貨の山。

^	罨 (ひとたび発見すれば表示される)。
)	武器。
[	服またはなんらかの防具。
%	食料 (衛生的であるとは限らない)。
?	巻物。
/	杖。
=	指輪。
!	水葉。
(	便利なアイテム (つるはし、鍵、ランプ…)
"	魔除けまたはくもの巣。
*	宝石または岩 (貴重なものかもしれないし、価値のないものかもしれない)。
~	岩、または彫像。
0	鉄球。
-	祭壇、または鉄の鎖。
{	泉
}	水たまり、または堀、または溶岩の海。
\	豪華な王座

a-zA-Z およびその他のシンボル これらの文字とその他のいくつかのシンボルは恐怖の迷宮のいろいろな住人たちを表す。彼らは不機嫌で悪意に満ちているかも知れないので警戒しなければならない。が、ときには何かの手助けになることもある。

I このマークは透明、あるいはその他の見えない怪物が最後にいたらしい場所に表示される。怪物はすでに移動しているかもしれないことに注意すること。ここでは 'F' と 'm' のコマンドが便利だろう。

これらのシンボルをすべて記憶する必要はない。 '/' コマンドでどのシンボルが何を表すか知ることができる (詳細は次の項を参照のこと)。

## 4 コマンド

コマンドは、そのコマンドに割り当てられている 1 文字か 2 文字のキー入力を行うか、拡張コマンドエントリのコマンド名を入力することで実行できる。いくつかのコマンド、例えば "search" はそれ以上 *NetHack* に情報を与える必要がない。他のコマンドは例えば方向や使用する物などの情報をさらに与える必要がある。これらさらなる情報を必要とするコマンドに対しては、*NetHack* は選択メニューもしくはコマンドラインプロンプトのいずれかを表示する。どちらが表示されるかは主として 'menustyle' オプションをどのように設定したかによる。

例えば "What do you want to use? [a-zA-Z ?\*]" という形式の質問がよくあるが、これはあなたに持っている物のうちどれを選ぶかを尋ねるものである。この質問において「a-zA-Z」はあなたが選べる持ち物の目録記号である。 '?' を入力するとこれらの物の目録一覧が得られ、個々の記号が何を表しているのか知ることができる。この例では、 '\*' もある。これは一覧には表示されておらず、通常そのコマンドでの使

用を想定していない物でも、使おうと思えばそれを選ぶことができることを示している。\*を入力するとすべての持ち物の目録が表示され、個々の持ち物すべてについての目録記号を知ることができる。考え直して結局このコマンドを使わないことにしたときには 'ESC' キーを押せばコマンドを終了することができる。

コマンドの中には、その前に数字を入力することで何回も繰り返すことができるものもある。例えば「10s」は 10 回の探索を表している。もし *number\_pad* オプションがオンのときには数字の前に 'n' を入力しなければならない。つまり上の例では代わりに「n 10s」と入力しなければならない。複数回実行することが無意味なコマンドではこれは無視される。さらに、移動コマンドの前に特定の文字 (プレフィックス) を付けることによってさまざまな移動方法をとることができる (後述)。繰り返し回数やプレフィックスを取り消すためには 'ESC' キーを押せばよい。

コマンドの一覧はかなり長い、ゲーム中に '?' コマンドを使用してヘルプメニューを見ることにより、いつでもコマンドの一覧を見ることができる。以下は標準キー配置の一覧である。

?	ヘルプメニュー: 表示可能なヘルプ画面のうちの 1 つを表示する。
/	これは何 コマンド; シンボルが何を意味するかを示す。指定するには場所を示すか、またはシンボル (あるいは単語全体) を入力するかを選択することができる。場所の指定は、適切な位置までカーソルを移動させてから '.'、','、';'、':' のどれかを押すことによって行なう。 '.' を押すと、選んだ場所で表示されているシンボルの説明が表示され、もし 'help' オプションがオンなら場合によっては "More info?" (詳細を見る?) と質問され、それからさらに他の場所を指定することができる。 ',' を押すと、シンボルの説明が表示されるが、詳細情報は表示されない; それから新しい場所を指定する。 ';' を押すと、詳細情報は表示されず、さらに他の場所の指定もできない。 ':' を押すと、(もしあるなら) 確認なしで詳細情報が表示される。場所を選ぶ時に ESC キーを押すことによってこのコマンドを中断でき、 '?' を押すとどのように動作するかの簡単な説明が表示される。 <i>autodescribe</i> がオンの場合、カーソルを動かす毎にその場所で見える物の短い説明が表示される。場所を選ぶ際に '#' を押すとこのオプションのオンオフが切り替わる。 <i>whatis_coord</i> オプションは短い説明にマップの座標を含むかどうかを制御する。場所ではなく名前を指定した場合は、常にその名前に対する追加情報が表示される。近くの怪物、現在表示されている全ての怪物、近くの物、全ての物の説明を要求することもできる。 <i>whatis_coord</i> オプションはその説明にマップの座標を含むかどうかを制御する。
&	コマンドの動作を示す。
<	上の階へ行く (階段やはしごにいるとき)。
>	下の階へ行く (階段やはしごにいるとき)。
[yuhjklbn]	指定した方向へ一歩進む (図 2 参照)。その方向に怪物がいると感じているか覚えているときは、移動するのではなく怪物と戦うことになる。怪物と戦うことになるのはこれらの一歩移動のコマンドだけである; その他のコマンド (後述) は「安全」である。

y k u	7 8 9
\   /	\   /
h- . -l	4- . -6
/   \	/   \
b j n	1 2 3

(*number\_pad* がオンなら)

図 2

- [YUHKLBN] 壁に突き当たるか何かに衝突するまで指定した方向に進む。
- m[yuhjklbn] プレフィックス: 物を拾ったり、(たとえそこに怪物がいると思っていても) 戦ったりせずに移動する。  
 いくつかの移動しないコマンドは、(一時的に *menustyle:Traditional* オプションを上書きして) メニュー経由での操作を要求するために 'm' 接頭辞を使う。一つの種類だけの物が存在している (“what kinds of objects?” と聞かれないので、ここで 'm' と答える機会がない) 場合に、' (拾う) を使うときに有用である。いくつかの他のコマンド (食料を食べる、生け贄を捧げる、缶詰を作る) では、'm' 接頭辞で床の上にある適用可能な物のチェックを飛ばして、すぐに持ち物のチェックに向かったり、(鞍を外すために “#loot” を使う場合) 入れ物を飛ばしてすぐに隣接する怪物に向かったりする。“#travel” コマンドではこの接頭辞で視界内の興味深い目標のメニューを表示する。デバッグモード (またの名を “ウィザードモード”) では、'm' 接頭辞は “#teleport” と “#wizlevelport” コマンドにも使える。
- F[yuhjklbn] プレフィックス: (そこに怪物がいると予想しているだけでも) 怪物と戦う。
- M[yuhjklbn] プレフィックス: 物を拾わずに遠くへ移動する。
- g[yuhjklbn] プレフィックス: 何かが見つかるまで移動する。
- G[yuhjklbn] or <CONTROL->[yuhjklbn] プレフィックス: 'g' と同じ。ただし通路の分岐点では止まらない。
- 指定された位置まで、最短距離アルゴリズムを用いて移動する。  
 最短距離は、あなたが知っている (見た、または以前に通った) 地図配置を基に計算される。知っている道がない場合、代わりに推測が行われる。'G' コマンドとほぼ同じ条件で停止するが、'M' コマンドと同様に物を拾わないで移動する。マウスに対応している版では、現在位置以外の位置をマウスでクリックしたときにもこのコマンドが発動する。
- . 待つ、または休憩する; 1 ターン何もしない。
- a 道具 (つるはし、鍵、ランプ…) を用いる (使う)。  
 杖に対して使うと、杖を壊し、その過程で魔力が開放される。確認が必要である。
- A 1 つまたは複数の身につけているもの (防具など) をはずす。  
 防具を 1 つだけはずすときには 'T'(take off: 防具をはずす) を、アクセサリを 1 つだけはずすときは 'R'(remove: アクセサリをはずす) を使用せよ。
- ~A 1 つ前のコマンドを繰り返す。
- c 扉を閉じる。

- C 怪物、個々のオブジェクト、オブジェクトの種類に名前を付ける。  
拡張コマンド “#name” と同じ。
- ^C パニックボタン。ゲームを放棄する。
- d 何かを下に置く。  
例えば d7a — a という物を 7 個下に置く。
- D いくつかの物を下に置く。  
次の質問  
“What kinds of things do you want to drop? [!%= BUCXaium]” (どの種類のものを置きますか?)  
に対して、0 個以上の物のシンボルを入力することで答える。さらに ‘a’, ‘i’, ‘m’, ‘u’ をその後に入力することもできる。加えて、祝福された (B)/呪われていない (U)/呪われた (C) アイテムを指定することもできる。  
DB — 祝福されていると判明している物全てを下に置く。  
DU — 呪われていないと判明している物全てを下に置く。  
DC — 呪われていると判明している物全てを下に置く。  
DX — 祝福/呪いが判明していない物全てを下に置く。  
Da — 確認なしにすべての物を下に置く。  
Di — 物を置く前に持ち物を確認する。  
Du — (店にいるとき) 代金未払の物だけを下に置く。  
Dm — 置く物を選ぶのにメニューを用いる。  
D%u — 代金未払の食料だけを下に置く。  
最後の例は組み合わせを示している。物のフィルタリングに三つのカテゴリがある: 種類 (水薬は ‘!’、巻物は ‘?’ など)、買い物の状態 (未払い、言い換えると店が持っているものは ‘u’)、祝福/呪い状態 (上述の通りの ‘B’, ‘U’, ‘C’, ‘X’)。一つのカテゴリで二つ以上指定した場合 (水薬と巻物で “!?” や、祝福と呪われていないで “BU” など)、持ち物は指定された値のどれかに一致した場合に条件を満たす (従って “!?” は ‘!’ または ‘?’ を意味する)。複数のカテゴリを指定すると、持ち物はそれぞれのクラスタの条件に一致した場合に条件を満たす (従って “%u” は種類 ‘%’ と未払い ‘u’ を意味する)。最後に、複数のカテゴリの複数の値を指定できる: “! ?BU” は、祝福または呪われていないと分かっている、全ての水薬と巻物を選ぶ。(3.6 より前のバージョンでは、フィルタの組み合わせは異なる振る舞いだった。)
- ^D 何か (普通は扉) を蹴る。
- e 食料を食べる。  
通常は床にある食べ物が調べられ、何も見つからないかどれも選ばなかった場合には持ち物の食べ物が調べられる。‘e’ の前に ‘m’ 接頭辞を置くことで床にあるものを食べようとするのを回避する。既に満腹のときに食べようすると、窒息して死ぬかもしれない。その危険を冒すなら、最初の一口を生き残ったなら “continue eating?” と聞かれる。返答に単なる ‘y’ ではなく “yes” が必要なように *paranoid\_confirmation:eating* オプションを設定できる。
- E 床にメッセージを刻み込む。  
E- — ほこりの中に指で書く。

- “Elbereth” という語を刻み込むと、たいていの怪物はあなたに白兵戦を挑んで来なくなる (しかしあなたが攻撃をすると文字は消えてしまう)。これは、一息いれたいときになかなか便利である。
- f 矢筒 (または矢筒袋または準備しているもの) に入れてあるものを発射する (投げるか撃つ)。発射するものは予め ‘Q’ コマンドで選択することも出来るし、*autoquiver* オプションが設定されている場合は、コンピューターに自動的に適当なものを設定させることもできる。  
より一般的に投げたり撃ったりするには ‘t’(投げる) も参照のこと。
- i 持ち物の目録 (持っている物すべて) を表示する。
- I 持ち物の目録のうち指定した一部を表示する; 普通は鎧なら ‘[’, 薬なら ‘!’ というようにオブジェクトの種類で指定する。  
I\* — 持ち物のうち宝石をすべて表示する。  
Iu — 代金未払いの物をすべて表示する。  
Ix — 代金未払いだが使ってしまった物をすべて表示する。  
IB — 祝福されていると分かっている物を全て表示する。  
IU — 呪われていないと分かっている物を全て表示する。  
IC — 呪われていると分かっている物を全て表示する。  
IX — 祝福/呪いの状態が分からないものを全て表示する。  
I\$ — お金を数える。
- o 扉を開ける。
- O オプションを設定する。  
現在のオプションの値の一覧が表示される。(見出し文字をタイプするかクリックすることによって) 変更したいオプションを選んで、ほとんどの値を変更することができる。ブール値でないオプションの場合は、さらにメニューが出るか、プロンプトが出る。設定可能なオプションはこのガイドブックの後ほどに一覧がある。オプションは通常は ‘O’ コマンドではなく、ゲームを始める前に設定する。後述のオプションの項を参照のこと。
- ^O 概要を表示する。“#overview” の短縮版: 訪れた興味深い階の一覧。
- P 代金を払う。
- P 装飾品 (指輪、魔除け、目隠し) を身につける。  
このコマンドは防具を身につけるのにも使える。どれを選ぶかの一覧では装飾品しか表示されないが、一覧に表示されていない防具を選択することでそれを身につけようとする事ができる。(後述する ‘W’ コマンドを参照のこと。これは一覧として防具が表示されるが、装飾品も選択できて、それを身につけようとする)。
- ^P 1 つ前のメッセージをもう一度表示する。  
続けて ^P を入力するとさらに前のメッセージが順に表示される。一部のインターフェースでは、この振る舞いは *msg\_windows* オプションによって変化する。
- q 何か (水薬、水など) を飲む。
- Q 矢筒、矢筒袋、あるいは単に一般的に準備するものを選択する (同時に指定できるのは

- これらのうち一つのみ)。ここで選択したものは ‘f’ コマンドで発射することができる。(バージョン 3.3 以前ではこのコマンドはゲームを放棄するものだったが、その機能は ‘#quit’ に移動した。)
- r 巻物や魔法書を読む。
- R 装飾品 (指輪、魔除け、目隠し) をはずす。  
 複数の物を身につけている場合、どれを外すかを指定する。一つしか身につけていない場合、標準設定では確認なしにその一つを外すが、確認が必要になるように *paranoid\_confirmation* オプションをセットできる。  
 このコマンドは防具を外すのにも使える。外す物を選ぶプロンプトには身につけている装飾品のみが表示されるが、身につけている防具も選ぶことができる。(後述の ‘T’ コマンドを参照のこと。これは持ち物の選択時に防具が表示できるが、装飾品も受け付けて外そうとする。)
- ^R 画面を描き直す。
- s 周囲の隠し扉や罠を探す。何かを見つけるには普通何回も探す必要がある。  
 隣接する「覚えているが見えていない怪物」マーカーの怪物がまだそこにいるかを確認するためにも使える。
- S ゲームをセーブする (プレイを中断してプログラムを終了する)。セーブされたゲームは、同じキャラクタ名を使った次回プレイ時に自動的に復旧される。  
 通常のプレイでは、セーブしたゲームが一旦復元されると、セーブしたデータを保存するのに使われていたファイルは削除される。探検モードでは、復元が完了した後ファイルを残すか削除するかを訊ねられる。ファイルを残すと、しばらくプレイした後セーブせずに終了し、後で再び復元することが可能になる。  
 例えセーブしたゲームファイルを残して再利用できる探検モードでも、“現在のゲームの状態をセーブしてプレイを続ける” コマンドはない。
- t 物を投げる。または矢などを発射する。  
 独立した “射撃” コマンドはない。弓を装備しているときに矢を “投げる” と、矢を撃つことになり、弓に関する武器スキルのボーナスやペナルティが適用される。弓を装備せずに矢を “投げる” と、手で矢を投げたことになり、一般的に撃つよりも効率的でない。  
 ‘Q’ (矢筒) コマンドで事前に選択された物を投げたり撃ったりする ‘f’ (射撃) も参照。
- T 防具をはずす。  
 複数の物を身につけている場合は、どれを外すかのプロンプトが出る。(例え実際に身につけていなくても、クロークは防具やシャツの上に身につけ、防具はシャツの上に身につけているものとして扱われる。) 一つしか身につけている場合は、デフォルトでは質問されることなくそれをはずすが、*paranoid\_confirmation* オプションを設定することでプロンプトを要求できるようにできる。  
 このコマンドはアクセサリをはずすのにも使える。どれを外すかを選ぶためのプロンプトには身につけている防具のみが表示されるが、身につけているアクセサリも選ぶことができる。(上述した ‘R’ を参照のこと。選択肢としてはアクセサリのみが表示されるが防具も選ぶことができ、それを外そうとする。)

^T	テレポート能力があればテレポートする。
v	バージョン番号を表示する。
V	このゲームの履歴を表示する。
w	武器を持つ。 w- — 武器として何も持たず素手 (またはグローブを付けた手) になる。 一部のキャラクタは二つの武器を同時に持つことが出来る; そうするには 'X' コマンド (または "#twoweapon" 拡張コマンド) を使う。
W	鎧を装備する。 このコマンドはアクセサリ (指輪、魔除け、目隠し) を身につけるのにも使われる。どれを選ぶかの一覧では防具しか表示されないが、一覧に表示されていないアクセサリを選択することでそれを身につけようとする事ができる。どれを身につけるかを選ぶためのプロンプトには防具のみが表示されるが、身につけるためのアクセサリも選ぶことができる。(前述した 'P' コマンドを参照のこと。選択肢としてはアクセサリのみが表示されるが防具も選ぶことができ、それを身につけようとする。)
x	装備している武器を予備の武器と交換する。 予備の武器は二刀流攻撃で用いられる。予備の武器がなくても交換は実行される (今持っているものを予備にして、素手になる) ことに注意。
X	キャラクタが実行可能なら、二刀流モードを切り替える。"#twoweapon" 拡張コマンドも利用可能である。 (3.6 より前のバージョンではこれは通常のプレイから「探索モード」またの名を「発見モード」に切り替えるコマンドであった; これは "#exploremode" に移動した。)
^X	あなたのキャラクタの基本的な情報を表示する。 名前、職業、(職業名によって冗長にならないなら) 性別、属性、従っている神および敵対している神を表示する。また、ステータス行の内容を、スペースの問題で通常のステータス表示には現れない追加のものも含めて、より詳細に表示する。 通常のプレイでは、これが '^X' が表示する全てである。探検モードでは、職業と状態は <i>enlightenment</i> (啓蒙) 魔法で提供される情報によって強化される。
z	杖を振る。 z. — 自分に振る場合は、方向に '.' を使う。
Z	呪文を唱える。 Z. — 自分を狙いたいときは、方向に '.' を使う。
^Z	ゲームを一時中止する (ジョブコントロール機能のある UNIX バージョンのみ)。
:	足元に何があるか見る。
;	見えているシンボルが何を示すかを表示する。
,	足下にあるものを拾う。 'm' の後に押すことによって、設定に関わらず選択メニューを表示する。
@	<i>autopickup</i> オプションのオン・オフを切り替える。
~	今までに発見した隣接した罫の種類を調べる。
)	持っている武器を表示する。

[	付けている防具を表示する。
=	はめている指輪を表示する。
"	付けている魔除けを表示する。
(	使っている道具を表示する。
*	装備しているものを表示する。 上の 5 つのコマンドを一つにしたもの。
\$	持っている金貨を数える。
+	知っている呪文の一覧を表示する。 このコマンドを使うと、リスト全体をソートするか、メニューから一つの呪文を選択して他の呪文と入れ替えることで、呪文が表示される順番を入れ替えることが出来る。呪文の組を入れ替えると詠唱の文字が変わるので、変更は今回の '+' コマンドが終了した後もそのままになる。リスト全体のソートは一時的なものである。今回の '+' コマンド以降も最近のソート順のままにするには、ソートオプションを再び選んで、“reassign casting letters” を行う。(これを行った後に覚えた呪文は、ソート順に挿入されるのではなく、リストの末尾に追加される。)
\	今までにどんな種類の物を見つけたかを表示する。
、	一つの種類の物の中で見つけた種類を表示する。
!	シェルに抜ける。
#	拡張コマンドを実行する。 以上から分かるように <i>NetHack</i> の作者たちはすべての文字を使い果たしてしまったので、あまり頻繁に使われないコマンドはこのようにして導入された。どの拡張コマンドが使用可能かは、ゲームのコンパイル時にどの機能が有効にされたかによる。
#adjust	持ち物の目録記号を変更する ( <i>fixinv</i> オプションが “on” の時に非常に便利である)。補完される。標準キーは ‘M-a’。 このコマンドは、ある特定の持ち物スロットから他のスロットに変更できる; これによりあなたにとってより意味のある文字にしたり、一覧表示時に特定の位置に現れるようにしたりできる。現在空いているスロットに移動させることもできるし、変更先が既に使われている場合—結合できないなら—そのいる物は位置が入れ替わる。“#adjust” はまた、物の塊を分割するのにも使える; 調整する物を選択するとき、文字の前に数を入力する。数を指定せずに調整する場合、以前は移動先に移すときに全ての互換性のあるまとまりは集められていた。この振る舞いは変更された; 互換性のあるまとまりを集めるには、まとまりを新しいスロットに “#adjust” する。名前が割り当てられている場合、その他の属性が同じなら同じ名前か名前のないまとまりが集められる。名前がない場合、その他の名前のないまとまりのみが有効である。どちらの場合も、異なる名前のまとまりは集められない。これはあるスロットから他のスロットに移すために “#adjust” を使った場合と対照的である。この場合、(数が指定されず) 互換性のあるまとまりを移動する場合には、どちらかに名前がついていてもう片方についていないなら、これは集められ、その名前が付けられる。一方、(数が指定されて) 分割される場合、移動先とまとまりを集めるかどうかを決めるときに元のまとまりの名前は無視される。

#annotate	現在の階層に結びつけられる 1 行の文を指定できる。注釈が付けられた全ての階層は “#overview” コマンドで表示される。補完される。標準キーは ‘M-A’ と、 <i>number_pad</i> がオンなら ‘^N’。
#apply	つるはし、鍵、ランプのような道具を用いる (使う)。標準キーは ‘a’。 道具が床にある物に使われる場合、‘m’ 接頭辞を使うとそれらの物は飛ばす。 杖に使うと、杖は壊れ、その過程で魔力が解放される。確認が必要。
#attributes	属性を表示する。標準キーは ‘^X’。
#autopickup	<i>autopickup</i> オプションを切り替える。標準キーは ‘@’。
#call	怪物や持ち物や床や発見物一覧にある物に名前を付けたり、現在の階に注釈を付ける (“#annotate” と同じ)。標準キーは ‘C’。
#cast	扉を閉じる。標準キーは ‘Z’。
#chat	誰かと話をする。標準キーは ‘M-c’。
#close	扉を閉じる。標準キーは ‘c’。
#conduct	維持している自発的挑戦の一覧を表示する。補完される。標準キーは ‘M-C’。詳しくは後述する “Conduct” の章を参照すること。
#dip	物を何かに浸す。補完される。標準キーは ‘M-d’。
#down	階段を降りる。標準キーは ‘>’。
#drop	物を下に置く。標準キーは ‘d’。
#droptype	特定の酒類の物を下に置く。標準キーは ‘D’。
#eat	何かを食べる。標準キーは ‘e’。‘m’ 接頭辞は床にある物を食べるのを飛ばす。
#engrave	床に何かを刻み込む。標準キーは ‘E’。
#enhance	武器の技量を高めたり、調べたりする。補完される。標準キーは ‘M-e’。
#exploremode	探検モードに入る。 確認が必要; 標準の返答は ‘n’ (ノー)。本当に探検モードに切り替えるなら、‘y’ と返答する。代わりに返答に “yes” を必要とするように <i>paranoid_confirmation:quit</i> オプションを設定できる。
#fire	矢筒から弾を発射する。標準キーは ‘f’。
#force	錠をこじ開ける。補完される。標準キーは ‘M-f’。
#glance	地図のシンボルが何を示すかを表示する。標準キーは ‘;’。
#help	ヘルプメニューを表示する。標準キーは ‘?’ および <i>number_pad</i> がオンの場合 ‘h’。
#herecmdmenu	現在の位置で可能な行動のメニューを表示する。
#history	長いバージョンとゲームの歴史を表示する。標準キーは ‘V’。
#inventory	持ち物の目録を表示する。標準キーは ‘i’。
#inventtype	指定した物の種類の一覧を表示する。標準キーは ‘I’。
#invoke	物が持つ特別な能力を発動する。補完される。標準キーは ‘M-i’。
#jump	別の場所へジャンプする。補完される。標準キーは ‘M-j’ および <i>number_pad</i> がオンの場合 ‘j’。

#kick	何かを蹴る。標準キーは ‘^D’ および <i>number_pad</i> がオンの場合 ‘k’。
#known	今までにどんな種類の物を見つけたかを表示する。標準キーは ‘\’。
#knownclass	一つの種類の物の中で見つけた種類を表示する。標準キーは ‘.’。
#levelchange	経験レベルを変更する。補完される。デバッグモードのみ。
#lightsources	携帯光源を表示する。補完される。デバッグモードのみ。
#look	足元に何かがあるか見る。標準キーは ‘:’。
#loot	あなたの足下の床に置いてある箱や鞆、またはあなたの隣に立っている馬などの鞍を漁る。補完される。‘m’ を前置すると現在の場所にある入れ物を飛ばして直接鞍を外す。標準キーは ‘M-1’ および <i>number_pad</i> がオンなら ‘1’。
#monster	(怪物の姿に変化している時に) 怪物の特殊能力を使う。補完される。標準キーは ‘M-m’。
#name	怪物、個々のオブジェクト、オブジェクトの種類に名前を付ける。“#call” と同じ。補完される。標準キーは ‘N’, ‘M-n’, ‘M-N’。
#offer	神にいけにえを捧げる。補完される。標準キーは ‘M-o’。 成功するためには祭壇を見つける必要がある。最近殺したモンスターの死体がよい選択肢である。‘m’ 接頭辞は、床にある物を捧げるのを飛ばす。
#open	扉を開ける。標準キーは ‘o’。
#options	オプション設定を確認、変更する。標準キーは ‘O’。
#overview	洞窟について発見した情報を表示する。(記憶喪失で忘れない限り) 全ての訪問済みの階と注釈が含まれ、多くのもの(祭壇、玉座、泉など; 他の洞窟へ分岐する追加の階段)があると自動的に注釈が追加される。ゲーム終了時の情報公開中に洞窟の概要が選ばれた場合、注釈のあるなしに関わらず全ての訪れた階層が含まれる。補完される。標準キーは ‘^O’ と ‘M-O’。
#panic	パニックルーチンをテストする。現在のゲームを終了する。補完される。デバッグモードのみ。 確認される; 標準設定は ‘n’ (ノー); プレイを続ける。本当にパニックさせるには、‘y’ と返答すること。代わりに “yes” との返答が必要にするために <i>paranoid_confirmation:quit</i> オプションを設定できる。
#pay	代金を支払う。標準キーは ‘p’。
#pickup	現在の位置にある物を拾う。標準キーは ‘,’。‘m’ 接頭辞はメニューの使用を強制する。
#polyself	自身を変化させる。補完される。デバッグモードのみ。
#pray	神に祈って助けを求める。補完される。標準キーは ‘M-p’。 前回の助けの後すぐに祈るのは良くない考えである。(ヒント: 洞窟に生きて入るということは助けを受けたものとして扱われる。新しいゲームを始めてすぐに祈るべきではないだろう。) このコマンドを間違えて使うのは問題を引き起こすので、祈る前に意思を確認するようにするオプションがある。これは標準設定ではオンになっており、オフにするために <i>paranoid_confirmation</i> をリセットすることが出来る。
#prevmsg	一つ前に表示されたゲームメッセージを表示する。標準キーは ‘^P’。

#puton	装飾品（指輪、魔除けなど）を身につける。標準キーは‘P’。
#quaff	何かを飲む。標準キーは‘q’。
#quit	ゲームをセーブせずにプログラムを終了する。補完される。標準キーは‘M-q’。 このコマンドを間違えて使うと現在のゲームを捨ててしまうことになるので、終了する前に意思を確認される。標準の返答は‘n’（ノー）；プレイを続ける。本当に終了するためには、‘y’と応答する。 <i>paranoid_confirmation:quit</i> オプションをオンにすることで、代わりに“yes”の入力が必要であるようにすることができる。
#quiver	矢筒のための弾を選択する。標準キーは‘Q’。
#read	巻物、魔法書、あるいはその他の何かを読む。標準キーは‘r’。
#redraw	画面を書き直す。標準キーは‘^R’および <i>number_pad</i> がオンなら‘^L’。
#remove	装飾品（指輪、魔除けなど）をはずす。標準キーは‘R’。
#ride	鞍の付いている生物に乗る（あるいは乗るのをやめる）。補完される。標準キーは‘M-R’。
#rub	ランプや石をこする。補完される。標準キーは‘M-r’。
#save	ゲームをセーブしてプログラムを終了する。標準キーは‘S’。
#search	周囲の隠し扉や罠を探す。標準キーは‘s’。
#seeall	装備している物を表示する。標準キーは‘*’。
#seeamulet	付けている魔除けを表示する。標準キーは‘”’。
#seearmor	付けている防具を表示する。標準キーは‘[’。
#seegold	持っている金貨を数える。標準キーは‘\$’。
#seenv	視界ベクターを表示する。補完される。デバッグモードのみ。
#seerings	はめている指輪を表示する。標準キーは‘=’。
#seespells	知っている呪文の一覧を表示し、並べ替える。標準キーは‘+’。
#seetools	使っている道具を表示する。標準キーは‘(’。
#setrap	隣接した罠の種類を表示する。標準キーは‘^’。
#seeweapon	持っている武器を表示する。標準キーは‘)’。
#shell	シェルに抜ける。標準キーは‘!’。
#sit	座る。補完される。標準キーは‘M-s’。
#stats	メモリ使用状況を表示する。補完される。デバッグモードのみ。
#suspend	ゲームを一時中止する。標準キーは‘^Z’。
#swap	装備している武器を予備の武器と交換する。標準キーは‘x’。
#takeoff	防具を一つ外す。標準キーは‘T’。
#takeoffall	全ての防具を外す。標準キーは‘A’。
#teleport	その階の中でテレポートする。標準キーは‘^T’。
#terrain	怪物、物、罠を表示せずに、基の地図を表示する。補完される。
#therecmdmenu	現在の隣の位置に対して可能な行動のメニューを表示する。
#throw	何かを投げる。標準キーは‘t’。

#timeout	タイムアウトキューを見る。補完される。デバッグモードのみ。
#tip	入れ物(鞆や箱)をひっくり返して中の物を出す。補完される。標準キーは‘M-T’。‘m’接頭辞はこのコマンドでメニューを使うようにする。
#travel	地図の指定された場所まで移動 (travel) する。標準キーは‘_’。“request menu”接頭辞を使うと、カーソル移動を訊ねることなく視界内の興味深い目標の一覧を表示する。カーソルで目標を選び、 <i>autodescribe</i> オプションがオンなら、もしあなたのキャラクタがその場所までの道のりを知らない場合には一番上の行に“(no travel path)”と表示される。
#turn	不死の怪物を追い払う。補完される。標準キーは‘M-t’。
#twoweapon	二刀流戦闘のオン・オフを切り替える。補完される。標準キーは‘X’、 <i>number_pad</i> がオフの場合は‘M-2’も使える。二刀流に適切な武器を使うこと。さもなければ自動的にオフになる。
#untrap	何か(罾、扉、宝箱)の罾をはずす。標準キーは‘M-u’と、 <i>number_pad</i> がオンなら‘u’。状況によっては罾に掛かった怪物を助けるのにも使えるかも知れない。
#up	階段を上る。標準キーは‘<’。
#vanquished	絶滅させた怪物の一覧を表示する。補完される。デバッグモードのみ。
#version	このバージョンの <i>NetHack</i> をコンパイルしたときのオプションを表示する。補完される。標準キーは‘M-v’。
#versionshort	バージョン文字列を表示する。標準キーは‘v’。
#vision	視界配列を表示する。補完される。デバッグモードのみ。
#wait	何もせずに1ターン休憩する。標準キーは‘.’と、 <i>rest_on_space</i> がオンなら‘ ’。
#wear	鎧を一つ装備する。標準キーは‘W’。
#whatdoes	あるキーが何をするかを示す。標準キーは‘&’。
#whatis	シンボルに対応している物が何かを表示する。標準キーは‘/’。
#wield	武器を装備する。標準キーは‘w’。
#wipe	顔を拭う。補完される。標準キーは‘M-w’。
#wizbury	あなたの下と周りに物を埋める。補完される。デバッグモードのみ。
#wizdetect	ある程度の範囲内に隠されたもの(隠し扉や罾や見えない怪物)を探す。補完される。デバッグモードのみ。標準キーは‘^E’。
#wizgenesis	怪物を造る。回数を前置することで複数造ることができる。補完される。デバッグモードのみ。標準キーは‘^G’。
#wizidentify	全ての持ち物を識別する。補完される。デバッグモードのみ。標準キーは‘^I’。
#wizintrinsic	いくつかの特性を設定する。補完される。デバッグモードのみ。
#wizlevelport	他の階に瞬間移動する。補完される。デバッグモードのみ。標準キーは‘^V’。
#wizmap	この階をマッピングする。補完される。デバッグモードのみ。標準キーは‘^F’。
#wizrumorcheck	噂の境界を検証する。補完される。デバッグモードのみ。
#wizsmell	怪物の匂いを嗅ぐ。補完される。デバッグモードのみ。

#wizwhere 特殊階の場所を表示する。補完される。デバッグモードのみ。  
 #wizwish 何かを願う。補完される。デバッグモードのみ。標準キーは '^W'。  
 #wmode 壁のモードを表示する。補完される。デバッグモードのみ。  
 #zap 杖を振る。標準キーは 'z'。  
 #? ヘルプメニュー: 利用可能な拡張コマンドの一覧を表示する。

もしあなたのキーボードにメタキー (別のキーと一緒に押すことによってそのキーの 'メタ'[第 8、または '上位'] ビットをセットする) があれば、コマンドの頭文字をメタキーと一緒に押すことによって多くの拡張コマンドを起動することができる。NT, OS/2, PC, ST NetHack では 'Alt' キーがこの目的に使われる。Amiga では、*altmeta* オプションを設定することでこの振る舞いが得られる。その他のシステムでは、'Alt' と他のキーをタイプすると *Escape* とタイプしたキーからなる 2 文字の並びが転送されるので、*NetHack* にこれを meta+キーと組み合わせるために *altmeta* オプションを設定できる。

M-? #? (対応していないプラットホームもある)  
 M-2 #twoweapon (*number\_pad* がオフの場合)  
 M-a #adjust  
 M-A #annotate  
 M-c #chat  
 M-C #conduct  
 M-d #dip  
 M-e #enhance  
 M-f #force  
 M-i #invoke  
 M-j #jump  
 M-l #loot  
 M-m #monster  
 M-n #name  
 M-o #offer  
 M-O #overview  
 M-p #pray  
 M-q #quit  
 M-r #rub  
 M-R #ride  
 M-s #sit  
 M-t #turn  
 M-T #tip  
 M-u #untrap  
 M-v #version

M-w                    #wipe

*number\_pad* オプションがオンのときは、これらに加えいくつかの文字コマンドが有効になる。

h                      #help

j                      #jump

k                      #kick

l                      #loot

N                      #name

u                      #untrap

## 5 部屋と通路

洞窟内の部屋や通路は明りがついていることもあるし、ついていないこともある。明りのついている部分で自分の視野に入る部分は画面に表示される。暗いところでは周囲 1 つ分の空間だけが見える。壁や通路は画面に表示されたままになる。

隠し通路は表示されない。これらは 's'(search: 探す) コマンドで発見することができる。

### 出入口

出入口は部屋と通路を接続するものである。出入口の中には扉のないものがある。このときにはそのまま通り抜けることができる。その他の出入口には扉があるが、その扉は開いているか、閉じているか、錠がかかっているかのいずれかである。閉じている扉を開けるには 'o'(open: 扉を開ける) コマンドを用いる。再び扉を閉じるには 'c'(close: 扉を閉じる) コマンドを用いる。

扉に錠がかかっているときは 'a'(apply: 道具を用いる) コマンドで錠をはずす道具を使うか、'^D'(kick: 蹴る) コマンドで扉を蹴破ることで通ることができる。

開いた扉に斜めから入ることはできない。水平あるいは垂直方向から真っ直ぐに近付かなければならない。扉のない出入口にはこのような制限はない。

扉は怪物を締め出すのに役に立つ。たいていの怪物は扉を開けることができない。けれどもいくつかの怪物にとっては扉を開ける必要などない (例: 亡霊は扉を通り抜けることができる)。

隠し扉は表示されない。これらは 's'(search: 探す) コマンドで発見することができる。一度発見すると普通の扉と同様になる。

### 罠 ('^')

そっかしい冒険者を陥れようとする罠が洞窟のあちこちにある。例えば落とし穴に落ちると上り出するのに数ターンの間その場から動けなくなるであろう。そこに誰かが足を踏み入れて引っかかるか、's'(search: 探す) コマンドで見つけるかして初めて罠は表示される。怪物も罠の餌食になることがある。これは非常に有効な防御戦術である。

古典的コンピュータゲームである“倉庫番”を基にした特別なダンジョンへの分かれ道も存在する。このダンジョンでの目的は大岩を落とし穴に入れることである。慎重に先読みすれば、全ての階は倉庫番の伝統の

ルールでクリアできる。行き詰まってしまった冒険者のためにいくつかの助け舟が用意されているが、これらは運を低下させてしまう。

## 階段とはしご (‘<’, ‘>’)

一般的に洞窟の各階には、前の階への上り階段 (‘i’) と、次の階への下り階段 (‘i’) がひとつずつある。しかし、例外もある。例えば、洞窟の比較的浅い階であなたは二つの下り階段がある階を発見するだろう。ひとつは洞窟の続きであり、もうひとつはノームの坑道として知られる地帯に続いている。この坑道は最後には行き止まりになっていて、ここを探検した (あなたがそうすることを選択した場合) 後、元の洞窟にまで戻って来る必要がある。

階段を使うか、あるいは他の階に移動する罠に引っ掛かった場合、いままでいた階は非活性化されて、ディスク上のファイルに貯えられる。昔訪れた階に再び訪れた場合、ディスク上のファイルが読み込まれて再活性化される。初めて訪れる階に移動した場合、その階は (ほとんどの階は一から、「特別な」階はテンプレートから、「骨の」階 (後述) の場合はファイルから読み込んで) 新たに作成される。怪物は現在の階にいるものだけが行動する。他の階にいるものは止まっている。普通、あなたが階段を使うと、あなたは目的地で対応する階段の上に現れる。しかし、ペット (後述) といくつかの怪物はあなたが階段を上り降りした時についてきて、時々それらの一体があなたと入れ替わることがある。こうなった場合、ペットやその他の怪物が階段の上に現れ、あなたはそのそばに現れる。

はしごは階段と同じ役目であり、階を繋ぐこの二つの種類はゲーム中ほとんど区別ができない。

## 店と買い物

時々ドアのそばに店主がいて、床にたくさんの物が置いてある部屋に行き当たることもある。物を拾って、‘p’ コマンドを使うことで物を買うことができる。物を拾う前に、物の上に立って “#chat” コマンドを使うことで値段を確認できる。支払いをする前に物を使うと借金になり、店主は債務を支払うまでは店の外に出してくれなくなる。

店の中にいるときに物を落すことで物を店主に売ることができる。売却価格を提示され、売るかどうかを確認されるか、単に店主が興味がないことを知らされる (一般的に、店で売っている物と同じ種類の物が買い取り対象である)。

偶然で物を店の中に落してしまうと、店主は普通代償なしに所有権を主張する。物を取り戻すためには買い戻す必要がある。

店主は資金不足になることがある。こうなった場合、何かを売ろうとすると現金の代わりに信用貸しを提案される。信用貸しは物を買うときに使えるが、同じ店でだけ有効である。他の店主は引き受けない。(迷宮で「クレジットカード」を手に入れることがあるが、これを店で使おうとしてはいけない。店主は受け入れない。)

\$ コマンドを使うと、今持ち運んでいる金貨の数 (かばんや箱に入っているものは除く) とともに、負債や信用貸しがあれば表示される。Iu コマンドは未払いの物 (まだ店が所有する物) の一覧 (あれば) を表示する。Ix コマンドは使ってしまった物の代金の一覧 (あれば) が表示される。

## 店の特殊な性質

店のいくつかの性質は意外なものかもしれない。

- アイテムの値段は色々な要素によって様々に変化する。
- 店主はドアの一つ内側の場所は店の外として扱う。
- 店主はあなたを鷹のように注意深く見張っているが、一般的にその他の消費者については無視する。
- 店が “closed for inventory” (「棚卸のため閉店」) の場合、待っていても開店はしない。
- 店は例えば品物が枯渇しても新しいアイテムが入荷することはない。

## 6 怪物

画面上にはあなたから見える怪物しか表示されないが、ひょっとすると闇の中で怪物とばったり出くわすことになるかもしれないので注意が必要である。魔法のアイテムの中には、怪物があなたを発見するよりも前にあなたが怪物を発見するのに役立つ物もある(しかしこの能力に大変優れている怪物もいる。)

‘/’ コマンドと ‘;’ コマンドが画面上に表示されている怪物に関する情報を得るのに用いられる。“#name” コマンド(標準では ‘c’ に割り当てられている)は怪物に名前をつけるのに用いられ、複数の怪物がいる時に、ある怪物を他のものと見分けたいときに便利である。名前としてスペースを指定すると、以前につけた名前を消すことになる。

拡張コマンド「#chat」によって、隣接する怪物と交流できる。実際の台詞は選択できない(言い換えると、あなたが何をしゃべるかは選べない)が、店主や Oracle of Delphi(デルファイの神殿)といった相手と話をすることは、有益な結果をもたらす。

### 戦闘

発見した怪物と戦いたいときには、単にその怪物に向かって移動するようにすればよい。多くの怪物はあなたが戦いを挑まない限り、他のことを気にすることはない。腹を立てると大変危険な怪物もいる。「三十六計逃げるにせず」ということわざを忘れずに。

ほとんどの場合、友好的な怪物の位置に移動することで攻撃をしようとする、意図を確認される。デフォルトでは ‘y’ を押すと確認したことになるが、移動に ‘y’ を使っていると間違いがちである。代わりに “yes” の返答が必要のように *paranoid\_confirmation* オプションを設定できる。

怪物が見えない(怪物が透明、またはあなたが目が見えない)場合、怪物がいるとあなたが思った位置に ‘I’ の文字が表示される。その区画に歩こうとすると、見えている時と同じようにあなたは怪物に攻撃しようとする。もちろん、もし怪物が移動してしまっていたら、あなたの攻撃は空を切ることになる。もし怪物がすでに移動してしまっていると考え、戦いたくない場合は、‘m’ コマンドを使うことで戦うことなく移動できる。逆に、そこに怪物がいるかどうかわからないけれどもとにかく戦って見たい場合は、‘F’ コマンドが使える。

### ペット

あなたは仔犬(‘d’)、子猫(‘f’)、ポニー(‘u’)のいずれかとともにゲームを始める。あなたのペットはあなたとともに洞窟をさまよい、怪物と戦う。ペットはあなたと同様に生きのびるための食料を必要とする。犬と猫は新鮮な死肉やその他の肉をえさにしている;馬は手に入れるのがより難しい草食の食料が必要であ

る。もしもペットのことが気がかりであったり、ペットを飼い慣らしておきたいと思うなら、あなたが直接えさを与えることもできる。そのためには食料をペットに向かって投げてやればよい。きちんと訓練されたペットはある種の状況のもとで極めて利用価値が高い。

ペットも怪物を倒していくにつれて経験を積んでいく。また時間とともに成長もし、体力や相手に与えるダメージも増加する。初めのうちはあなたよりもペットの方が強いだろうから、レベルの低いキャラクタにとっては役に立つだろう。

あなたが他の階へ移動するとき、隣にペットを連れていけばペットもあなたについて移動する。置き去りにすると野生化してしまうかもしれない。同様に、もしあなたが自分の位置が変わるような罠にかかった場合(例えば、トラップドアで下の階に落ちた場合)、隣にいたペットはついていき、隣にいなかったペットは取り残される。ペットが自分でそのような罠にかかることもある。あなたはたとえその時にペットの隣にいても一緒に移動することはない。

## 軍馬

ダンジョンにいるある種の生物は、あなたが正しい装備と能力を持っていれば、乗ることができる。野生の生物にあなたが乗ることを納得させるのは困難である。多くの洞窟の主は協力関係をでっち上げるのに魔法の力に頼らなければならない。しかし、一旦動物をあなたの支配下に置いたなら、“#ride” コマンドで乗り降りすることができる。動物に乗って洞窟を移動する方法は、自分自身が移動するのと同じである。地図に表示されるのはあなたが乗っている動物である。

乗馬技術は‘#enhance’ コマンドで管理することができる。詳しくは後述する武器の技術の章を参照すること。

鞍を隣にいる生物に付けようとするには、‘a’ (apply) コマンドを使って持物から鞍を選ぶ。成功すると、それはその生物の持物に移される。

隣にいる鞍の付いている生物から鞍を外そうとするには、“#loot” コマンドを使う。成功すると、それはあなたの持物に移される。

## 骨の階

冒険者の幽霊とその死体(あるいはそれは生まれ変わる前のあなた自身であることさえある)や、その所持品に出くわすかもしれない。幽霊を殺すことは難しいが、動きがのろくほとんどダメージを受けないので簡単に逃げるができる。あなたは死んだ冒険者の所持品を奪うこともできる。しかしそれらは呪われていることが多い。以前の冒険者を死に到らしめたものにはすべて注意を払わねばならない。それはあたりを徘徊し、その直前の勝利に酔いしれているからだ。

## 怪物の永続性

怪物(一般に人間とペットを含む表現)は、それが見えるかその他の方法で感知できる場合にのみ表示される。怪物があなたが見えなかったり感知できない場所に移動すると、それがあなた以外の誰かの場合と同様、地図から消える。

しかし、見えたり感知できたりしない怪物と遭遇した-おそらく見えず、出会い頭に出会った-場合、特別な「覚えているが見えない怪物」マークが、それがいるとあなたが考えている場所に表示される。これは、

見え見えない怪物が他の場所に移動していたり、あなたが通常マーカの位置が見えない位置に移動したとしても、あなたがそこに怪物がいないと確信するまで表示され続ける。

## 7 物

洞窟の中で何かを見つけた場合、それを拾いたいと思うのはよくあることである。*NetHack* ではその物の上を通ることによって自動的に拾うことができ (*autopickup* オプション (後述) がオフになっているときや `'m'` プレフィックス (前述) を用いて移動するときはこの限りではない)、または `';` コマンドを使って手動で拾うことができる。持ち物が多すぎるときには、*NetHack* はあなたにそのように告げ、それ以上拾うことはできなくなる。そうでなければ *NetHack* はその物をあなたの荷物に加え、何を拾ったかが表示される。持ち物が増えるに連れて、荷物の重さが重くなる。どれだけの重さの物を運べるかは筋力と耐久力による。力が強く丈夫なほど、荷物の重さの影響は小さくなる。しかしそれでも、洞窟の中を持ち歩いている荷物の重さがあなたに負担になるときがある。反応は鈍くなり、カロリー消費が早くなってより多くの食料が必要となる。そして最終的には荷物が重すぎて何かを捨てないと動けなくなってしまふ。*NetHack* では荷物の重さがどれくらい悪影響を与えるかを教えてくれる。荷物が邪魔になってきた場合、状態 *“Burdened”*, *“Stressed”*, *“Strained”*, *“Overtaxed”*, *“Overloaded”* がの一つが最下行に表示される。

物を拾ったとき、その物には目録記号が割り当てられる。物に関する多くのコマンドはあなたがどの物を使いたいのかを尋ねてくる。持ち物のうち特定の物を選ぶように *NetHack* が尋ねてきたときは、普通目録記号の一覧が表示されてその中から選ぶ (前述のコマンドの項を参照のこと)。

いくつかの物、例えば武器類のような物は、どんなものか簡単に区別が付けられる。この他の物、例えば巻物や水菓はその種類に応じていろいろな名前が付けられている。1 回のゲームの間は同じ名前を持った物は同じ種類の物である。しかし物に付けられる名前はゲームごとに違うものになる。

このような物を使った時、もしその効果が明らかな場合は *NetHack* がそれが何であるかを憶えてくれる。その効果があまり明かでない場合はその物の種類を何と名付けるか尋ねてくる。このため後になってそれを思い出すことができる。またいつでも `“#name”` コマンドを使って同様のことができ、ある種類の物すべてに名前を付けたり個々の物に名前を付けたりできる。すでに名前をつけているものに `“#name”` を使う場合、新しい名前としてスペースを指定することによって、以前つけていた名前を消すことができる。

## 呪いと祝福

あなたが発見したいろいろな物は、たとえそれが役に立つものであったとしても呪いがかけられているかも知れない。呪いがかけられている物を使うとそれが張り付いてしまうことは最もよく見られる結果である。呪いがかけられている武器を持つとそれは手に張り付いてしまって取れなくなる。呪いがかけられている物を身につけると普通の方法でははずすことができない。さらに、呪いがかけられている武器や防具には必ずと言うわけではないがたいていの場合負の魔力が与えられていて、そうでない物よりも戦闘時の効力が劣る。その他の呪いがかけられている物は、あまり役に立たなかったりまたはその他の点で害を及ぼしたりするだろう。

また祝福されている物もある。祝福されている物は呪いがかけられていない普通の物に比べて具合良く働き役に立つ。例えば祝福された武器は悪魔たちに一層のダメージを与えることができるだろう。

呪われてるわけでも祝福されているわけでもない物は「呪われていない (uncursed)」と呼ばれる。これは「祝福されていない (unblessed)」と表現することもできるが、ゲームの中では「呪われていない」とい

う表現が使われる。これは「コップに半分だけ入っている/半分しか入っていない」問題である; 解釈はあなた次第である。

魔法を使うと物に呪いをかけたり呪いを解いたりすることができる。このためたとえ物が張り付いてしまっても、呪いを解いてははずすことができる。僧侶は生来呪いや祝福に敏感なので、他の職業の冒険者よりも容易に呪われているものを避けることができる。

あなたの持ち物の目録の中で呪いがかけられているかどうか分からない物は目録のなかで特に何の説明もない。どんな状態にあるか知っている物は目録一覧で「cursed(呪われた)」「uncursed(呪われていない)」「blessed(祝福された)」などの言葉が物の説明に与えられる。場合によっては、その他の情報が十分に表示されている場合は「uncursed(呪われていない)」は省略される。*implicit\_uncursed* オプションはこれを制御するのに使われる; これをオフにすると、例えその他の属性から推測される場合でも「uncursed(呪われていない)」を表示する。

## 武器 ('')

恐怖の迷宮に住むほとんどすべての怪物は、チャンスと見れば見境なくあなたを殺そうとするだろう。自分の身を守る(先に怪物を殺してしまう)ためにあなたは武器を必要とする。武器なしではポイントにして1~2のダメージ(いくらかの加算があるかも知れないが)を与えることしかできない。但し、モンクは例外である。モンクは武器を使って攻撃するよりも素手(あるいはグローブをつけた手)で攻撃した方が大きなダメージを与えることができる。

武器には鎚矛や剣のような振り回すためのものと、矢や槍のような投げつけるためのものがある。武器で怪物に打撃を与えるためには、武器を手にもって怪物を攻撃するか武器を怪物に投げなければならない。投げるには、単純に槍を投げるように選択するだけでよい。弓から矢を発射するには、まず弓を手にとってから矢を投げればよい。クロスボウは crossbow bolt(クロスボウ用の太矢)を発射するためのものである。投石器は岩や(宝石のような)その他の石を投げつけるために用いる。

魔力のある武器には「追加能力」(攻撃能力と言うことで正と負の両方を取り得る)が付けられており、攻撃が怪物に当たる可能性と与えるダメージに追加される。武器の魔力を測定するには、なんらかの方法で魔法を使って鑑定するしかない。多くの武器は錆のようなある種のダメージを受けやすい。このような「腐食」ダメージは修復可能である。

攻撃がうまく怪物に命中するかと、命中した時にどれくらいのダメージを与えるかは、多くの要素によって決定される。それらには以下のものがある: 武器の種類、武器の品質(魔力や腐食)、経験レベル、筋力、敏捷性、荷物の重さ、技量(後述)。怪物の防御値(一般的な防御率。防具を着ることによるとは限らない)も要素の一つである。また、ある種の武器に対して特に耐性を持つ怪物もいる。

多くの武器は片手持ちであるが、両手が必要な武器もある。両手持ちの武器と盾を同時に持つことはできない。片手持ちの武器を持っている場合、もうひとつの武器を予備として準備しておいて、'x' コマンドで今使っている武器と交換することができる。さらに「二刀流戦闘」技能の技量がある場合、二つの武器を同時に使うことができる。'x' コマンドで二刀流戦闘をするかどうかを切り替えられる。一部のキャラクター(例えば野蛮人)だけがこの技能を持っている。たとえこの技能を持っていても、一度に二つの武器を使うとひとつしか使わない時に比べて敵に命中する確率にペナルティを受ける。

全く何の武器も装備したくない時もあるかもしれない。そうするためには、装備するものを選ぶ時に '-' を選ぶか、あるいは 'A' コマンドでその他の身につけているものを外すのと同時に現在使っている武器を外すことができる。

AD&D をプレイしたことがある読者はお気づきだろうが、AD&D に登場した武器は *NetHack* においても怪物にだいたい同じダメージを与える。あまり良く知られていない武器 (*aklys*(アキリス), *lucern hammer*(ルツェンハンマー), *bec-de-corbin*(ベッグ・デ・コルビン) など) のいくつかは AD&D の追加ルールである *Unearthed Arcana* の付録で詳しく説明されている。

武器を使うコマンドは ‘w’(wield: 武器を持つ)、‘t’(throw: 投げる)、‘f’(fire: 武器を発射するもうひとつの方法)、‘q’(quiver: 矢筒)、‘x’(exchange: 武器を交換する)、‘X’(二刀流)、‘#enhance’(後述) である。

## ものを投げることと発射すること

‘t’ コマンドを使うことによって、どんな物でも投げることができる。このコマンドを指定すると、投げる物を探ねてくる。‘?’ を押すと、持ち物の中で投げるのに向いていそうな物の一覧が表示され、‘\*’ を押すと、全ての持ち物の一覧が表示される。何を投げるかを指定した後、(どの目標に投げるかではなく) どの方向に投げるかを尋ねられる。投げることができる距離は、主に物の種類と筋力による。矢は手で投げることでもできるが、弓を持って投げた方が遥かに遠くまで飛び、目標にも当たりやすい。

‘q’ コマンドで予め好みの「発射物」を選択しておき、‘f’ コマンドで投げることによって、投げる操作を簡単にすることが出来る。先程と同様に投げる方向は尋ねられるが、‘f’ コマンドを使うたびに何を投げるかを指定する必要はない。*autoquiver* オプションをオンにすると、‘q’ コマンドで設定した物がなくなった時に、*NetHack* が自動的に矢筒(または矢筒袋または準備している物)に他の物を入れてくれる。

一度に複数の物を投げることができるキャラクターもいる。一度に複数の矢を装填し(あるいは一度に複数の物を持ち)、それを目標に当てることはたやすい仕事ではない。レンジャーはこの技能に優れているし、適切な武器の技能(弓を持って矢を射るなら弓の技能、クロスボウを使うならクロスボウの技能、スリングを持って石を投げるならスリングの技能)において高いレベルの技量があれば他の職業でも可能である。あなたがいくつのアイテムを同時に投げることができるかは毎ターン変化する。‘t’ コマンドや ‘f’ コマンドの前に数字を指定することによって、発射する数を明示的に制限することもできる。例えば、“2f”( *number\_pad* オプションがオンの時には “n2f”) と指定すると、たとえ 3 本発射可能でも最大 2 本しか発射されない。もし、実際に発射できる数よりも大きい数を指定した場合(この例では “4f”) は、特に数字を指定しなかった場合と同様、実際に発射可能な数(ここでは 3 本)しか発射されない。一旦動作を開始したなら、全てのアイテムは同じ方向に飛ぶ。もし 1 本目で怪物を倒したなら、残りはその先まで飛んでいくことになる。

## 武器の技量

利用できる武器の技能はさまざまに異なっている。武器の技量(技能)はある種の武器をどれくらいうまく扱えるかに影響する。技能はゲームを通じて向上させることが出来るが、それは職業や経験レベルや武器の使用回数による。

技量を示すために、武器は *daggers*、*broadwords*、*polearms* といった風にいくつかのグループに分けられている。それぞれのグループで技量レベルをどこまで高めることが出来るかは職業毎に決まっている。例えば魔法使いは *daggers* や *staves* に対しては高いレベルに達することが出来るが、*swords* や *bows* に対してはそうではない。

‘#enhance’ 拡張コマンドで現在の武器(と呪文)の技量を見ることが出来る。さらに、ひとつまたは複数の技能を向上させることが出来る状態なら、どの技能を向上させるかを選択することが出来る。技能のランクは “none”(技能を向上させることができないという意味で “restricted”(制限された)と呼ばれるこ

ともある)、“unskilled”(初心者)、“basic”(入門者)、“skilled”(熟練者)、“expert”(エキスパート)である。制限された技能は単に“#enhance”コマンドで一覧に表示されない。(神の介入によりある種の技能の制限が解除されることもある。その時には技能は“unskilled”となり、限界は“basic”までとなる)キャラクターによっては、素手での戦闘やマーシャルアーツの技能を“master”(マスター)や“grand master”(グランドマスター)にまで向上させることができる。

技能レベルが“restricted”や“unskilled”の武器を使うと、怪物への命中率や命中した時に与えるダメージにペナルティがある。“basic”ではペナルティもボーナスもない。“skilled”では命中率と与えるダメージにささやかなボーナスがある。“expert”ではボーナスは大きくなる。攻撃が命中すると、使っている武器に関する技能を(もし最大に達していないなら)向上させるための訓練度を増やすチャンスがある。この訓練度が次のレベルに必要な値に達すると、自分の技能により自信を持てるように感じたことが告げられる。この時点で“#enhance”コマンドを使うことによってひとつまたは複数の技能を向上させることができる。技能は自動的に向上しない。なぜなら全ての技能レベルの合計には制限があるので、どの技能を向上させて、どれを無視するかを決める必要があるからである。

## 二刀流

一部のキャラクタは二つの武器を一度に使うことができる。そうするための準備は厄介だが、使うことで身についていく。二つの武器を装備するには、“#twoweapon”コマンドを使う必要がある。しかし、まずそれぞれの手に一つの武器を手にする必要がある。(二つの武器は完全に同じではないことに注意すること;通常装備している手で手にしている武器は主要武器として扱われ、もう一方は補助武器として扱われる。もっとも注意すべき違いは、二つの武器を手にするのを止めた後—または始める前—である。優先武器は手にしている武器であり、補助武器は単に、予備の武器として指定された、持ち物の一つである。)

主要武器を手にしていてももう片方の手が空いていたり間違った武器を持っていたりする場合、‘x’, ‘w’, ‘x’ とすると、まず主要武器をもう片方の手に持ち替え、補助武器として使いたいものを装備し、再び意図通りの手に入れ替えられる。補助武器が正しいけれども主要武器が間違っている場合、単に‘w’を使って主要武器を手にする事ができる。最後に、どちらの手にも正しい武器を手にしていない場合、‘w’, ‘x’, ‘w’ として、最初に補助武器を手にし、それを入れ替え、主要武器を手にする。

処理全体は *pushweapon* オプションを使うことで単純化できる。これを有効にして、何かを装備するのに‘w’を使うと、現在装備している武器を予備の武器にする。従って‘w’, ‘w’ と続けると、最初に予備にした武器を装備し、それから優先したい武器を装備すると、最初の武器が予備の武器に押し出される。

二刀流モードのとき、‘x’ コマンドを使うと、単一武器モードに戻る。どちらかの武器を投げたり落としたりするか、どちらかが盗まれたり壊されたりしても、単一武器モードに戻る。

## 防具 (‘[’)

多くの非友好的なものが洞窟には潜んでいる。そういったものからの攻撃から我が身を守るには防具が必要である。防具の中には他の種類の防具よりも防御効果に優れた物がある。Armor class(AC)(防御値)はこの防御効果の尺度である。防御値(AC)はAD&Dの場合と同じように評価される。10が防具なしの状態と等しく、数値が小さいほど優れた防具であることを示す。AD&Dに存在する鎧はいずれも *NetHack* においても同じ防御効果を示す。以下に(不完全ではあるが)各種の鎧ごとに規定される防御値の一覧を示す。

dragon scale mail(ドラゴンの鱗鎧)	1	plate mail(鋼鉄の鎧)	3
crystal plate mail(水晶の鎧)	3	bronze plate mail(青銅の鎧)	4
splint mail(鉄片の鎧)	4	banded mail(帯金の鎧)	4
dwarvish mithril-coat(ドワーフのミスリル服)	4	elven mithril-coat(エルフのミスリル服)	5
chain mail(鎖かたびら)	5	orcish chain mail(オークの鎖かたびら)	6
scale mail(鱗の鎧)	6	dragon scales (ドラゴンの鱗)	7
studded leather armor(釘付き皮鎧)	7	ring mail(鉄環の鎧)	7
orcish ring mail(オークの鉄環の鎧)	8	leather armor(皮鎧)	8
leather jacket(皮ジャケット)	9	no armor(鎧なし)	10

さらに他の防具 (例: 兜、靴、楯、クローク) を身につけ、防御値の値を小さくすることもできる。ただし同じ範疇に入る物は 1 つしか身につけることはできない (鎧は一式、クロークは 1 着、兜は 1 個、楯は 1 枚など)。

防具に魔力があればその防具の防御効果は通常のものよりも良く (もしくは悪く) なっており、それに付いている「+記号」(あるいは-記号) の分だけ防御値の値が小さくなる。例えば +1 鎖かたびらは、通常の鎖かたびらよりも防御効果が高く、防御値は 1 単位分小さくなって 4 になる。防具を身につけると直ちに防御値と「+記号」の値が分かる。呪いがかけられた防具は通常負の魔力 (-記号) を持っており、取りはずすことはできない。

多くの鎧は錆のようなある種のダメージを受けやすい。このようなダメージは修復可能である。魔法を唱えるのを妨げる防具もある。

鎧を使うためのコマンドは 'W'(wear: 防具を付ける) と 'T'(take off: 防具をはずす) である。'A'(remove all: 全てをはずす) もまた他の身につけるもの同様に、防具をはずすのに使用できる。

## 食料 ('%')

生きのびるには食料が不可欠である。長い間食料を口にしないまましているとやがて昏倒し、徐々に餓死への道をたどることになる。保存の処置を取ってないと傷んでしまい、食べるには不衛生になる食料もある。アイスボックスや缶に入っている食料は通常はいつまでも傷まないが、アイスボックスは重く、缶は開けるのに少しばかり時間が必要である。

怪物を殺すと通常はその死体が残るが、これは「食料」にもなる。すべてと言うわけではないが多くの食べられるし、中には食べると特別な力がつくものもある。ざっくり言えば「食べたものになる」ということである。

肉食主義の職業や怪物もいる。肉食主義の怪物は決して動物の死体を食べない。一方肉食主義のプレイヤーは動物の死体を食べることができるが、なんらかのありがたない副作用がある。

*fruit* オプションによってあなたの好物の食料にちなんでその名前を 1 つ設定することができる。

食料を食べるためのコマンドは 'e' である。

## 巻物 ('?')

巻物にはいろいろな表題が付けられているが、これはおそらくいしえの魔術師たちが暇潰しとして選んだのであろう (例: "READ ME" とか "THANX MAUD" の逆さ読みとか)。巻物は読むと消滅する (ただし魔法の呪文の書かれていない白紙は消滅しない)。

これらの中で最も利用価値の高いものは *scroll of identify*(識別の巻物) である。これは他の物についてそれが何であるか、呪いがかけられているか祝福されているか、あと何回効力を発揮できるかを確定できる。得体の知れない魔力を持つ物の中にはこの巻物なしでは何であるか鑑定し難いものもある。

メイルデーモンが走ってきてあなたに *scroll of mail*(手紙の巻物) を渡してくれることもある(この機能を有効にしてコンパイルされているバージョンの場合)。NetHack のメール配達機能はシステムのメールボックスに電子メールがやってきた場合に起動される。この機能を使用するには環境変数“MAIL”にあなたのメールボックスのファイル名を設定して NetHack に新しいメールを探す場所を知らせる必要がある。また望むならば環境変数“MAILREADER”に使いたいメール受信プログラムのファイル名を設定することもできる。このとき NetHack からそのプログラムを起動してその巻物を読むことができる。メールがゲーム内においてランダムに生成されるバージョンの NetHack ではこれらの環境変数は無視される。メイルデーモンは *mail* オプションでオフにできる。

巻物を読むためのコマンドは‘r’である。

## 水薬 (‘!’)

水薬は小びんに入っている液体の色によって区別される。水薬は飲むと消滅してしまう。

透明な水薬は水である。これらはしばしば祝福されていたり呪いがかけられていたりして、聖水や不浄な水になったりする。不死の怪物にとって聖水は有害なので、聖水を不死の怪物に投げつける(‘t’)と有効である。聖水に他の物を浸す(“#dip”)のもたいへん有益である。

水薬を飲むためのコマンドは‘q’(quaff: 飲む)である。

## 杖 (‘/’)

杖は通常複数回魔法効果を発揮する。一部の杖は杖を振る際に方向を指示する必要がある。杖を自分に向けて振ることもできる(方向として‘.’か‘s’を入力する)が、これはしばしば愚かな行為となる。その他の杖は方向は必要でない。杖が効力を発揮する回数は杖ごとに不定で、杖を使う度にその回数は1ずつ減る。

杖の魔法の量がなくなると、通常は杖を使用しても何も起きない。しかしながら時折、最後の一握りの魔力を使いきった杖から搾り取ることが可能である。しかしこれをするとう杖は壊れてしまう。杖は適切な魔法によって再充填することができるが、そうすると杖が爆発してしまう可能性がある。爆発する可能性は最初は非常に小さく、同じ杖に何度も再充填するごとに大きくなる。

自暴自棄な行為ではあるが、どうにもならなくなったときには杖を壊すという手もある。これは弱気な行為ではない。それを行うことにより、破壊的な魔法のエネルギーが解放されるからである。

ある杖を完全に鑑定した場合、持ち物の一覧では括弧の中に追加の情報が表示される。それは再充填された回数、コロン、現在の使用可能な回数である。杖が無効化された場合には使用可能な回数は-1になる。

杖を使うためのコマンドは‘z’(zap: 杖を振る)である。壊すには‘a’(apply: 道具を使う)である。

## 指輪 (‘=’)

指輪は大変役に立つ物である。というのも水薬や巻物や杖のように魔力が一過性にしか働かないものと違い、指輪の魔力は比較的恒久的であるからだ。

指輪をはめることによってその魔力は発揮される。指輪は両手の薬指に1つずつ、計2つしかはめることはできない。

またたいの指輪は身につけると腹の減り方が速くなる。減り方は指輪の種類によって異なる。指輪を使うためのコマンドは 'P'(put on: 指輪をはめる) と 'R'(remove: 指輪をはずす) である。

## 魔法書 ('+')

魔法書は強力な魔法を記した大きな本である。'r'(read: 読む) コマンドで学ぶと呪文の知識が読み手に転送される (したがって最後には読めなくなる) か、さもなくばその試みは不測の結果に終る。呪いがかけられた魔法書や知識の及ばない神秘の古代文字で記された魔法書を読むと健康状態に害が及ぶ可能性がある!

呪文を唱えたとき (学んでいる時も) にも魔力が逆流することもある。あなたの経験レベルでは遠く及ばない高度な呪文を唱えようと試みたり、対応する呪文タイプに関する技能が低かったり、ひどくついていないときに呪文を唱えたりすると、呪文を唱えるのに必要なエネルギーと時間を浪費しただけに終わってしまうこともある。

呪文を唱えることは、魔法のエネルギーを呼び起こしてそれを精神そのものに集中させることである。あなた自身の内部から出る魔法のエネルギーが開放される。唱えると一時的に魔法の力が吸い出される; これはゆっくりと回復し、その際に追加の食料が必要になる。呪文を唱えるには練習が必要である。練習していれば、それぞれの呪文領域の技能は向上していく。しかしながら、時間が経つにつれてそれぞれの呪文に関する記憶は薄れ、もう一度呪文を学ぶ必要がでてくるだろう。

一部の呪文は、杖と同様、呪文を唱える方向を指定する必要がある。自分自身に対して唱えるには、方向として '.' または 's' を指定する。いくつかの呪文は単なる方向ではなく場所を指定する必要がある。その他の呪文は方向や場所を指定する必要はない。

武器が習熟度の面でグループ分けされているのと同様、呪文もグループ分けされている。呪文を唱えるのに成功すると、そのグループを訓練したことになる;十分に訓練したスキルを向上させる '#enhance' コマンドはそのグループの全ての呪文に影響する。スキルが向上すると呪文の性能が向上し、詠唱時の失敗のリスクが減り、どれくらい長い間呪文が記憶に残っているかの推定の精度が向上する。技能スロットは武器スキルと共有する。(“武器の技量”の項も参照のこと)

魔法を唱えるには自由に移動できなければならない。また、いくつかの種類の鎧を着ていると魔法を唱えるのを妨げるだろう。

魔法書を読むためのコマンドは巻物を読むときと同じく 'r'(read: 読む) である。 '+' コマンドにより、知っている呪文毎に、レベル、スキル分野、成功率、どれくらい強く記憶しているかが一覧表示される。 'Z'(cast: 呪文を唱える) コマンドにより呪文を唱える。

## 道具 ('()')

道具はいろいろな目的に使う種々雑多な物である。杖などと同じように使用回数に制限のある物もある。例えばランプはしばらくたつと燃え尽きてしまう。その他の道具の中には容器も含まれており、物を出し入れすることができる。

道具を使うためのコマンドは 'a'(apply: 道具を用いる) である。

## 箱

冒険の途中で鞆や箱やひつに出くわすこともあるであろう。これらは置いてある場所に立っている場合には拡張コマンド '#loot' によって、また持っているときには 'a'(apply: 道具を用いる) コマンドで開け

ることができる。しかしながらひつにはしばしば錠がかかっており、大抵は重くて運びにくい物体である。ひつは手に持つことはできない物なので、錠をはずす(‘a’(apply: 道具を用いる) コマンドで鍵や錠をはずす道具を使ったり、‘D’ コマンドで蹴とばしたり、拡張コマンド “#force” により武器を使ってこじ開けるなどの方法による) には床に置かなくてはならない。

ひつには罠が仕掛けられているものもあり、錠をはずしたりふたを開けたりしたときに不快な出来事が起きる。拡張コマンド “#untrap” によってチェックをして罠が無効になるよう試みることができる。

## 魔除け (‘”’)

魔除けは指輪と大変よく似ており、しばしばもっと大きな効力を持っている。指輪と同じように魔除けにはいろいろな魔法の特性があり、有益なものも有害なものもあって、身につけることにより効力を発揮する。

魔除けは一つだけしか首にかけることができない。

魔除けを使うためのコマンドは指輪の場合と同じく ‘P’(put on: 指輪をはめる) と ‘R’(remove: 指輪をはずす) である。

## 宝石 (‘\*’)

宝石の中には価値のあるものもあり、高値で売れる。宝石は財産を持ち歩く方法としてとして極めて効果的なやり方である。価値のある宝石を脱出時に所持していれば得点に加算される。

他の小さな石も宝石に分類されるが、その価値はほとんどない。しかし、全ての石は(もしスリングを持っているなら) 弾として有効である。窮余に陥いたら手で投げることも可能である。

## 大きな岩 (‘”’)

彫像や巨石は特に有用ではないし、一般的に重いものである。見かけとは異なった彫像もあるという噂である。

巨大なヒューマノイド(巨人やその仲間)は巨石を武器として用いることができる。

一部のプログラム設定では、彫像はもはや‘”’ではなく、表現されている怪物を意味する文字で表される。

## 金 (‘\$’)

金は得点に加算され、また店では金で物を買うことができる。迷宮にはあなたの持つお金に影響されるかもしれない数多くの怪物がいる(店主は別として)。

## 物の永続性

通常、あなたが地図上のある特定の位置である物を見て、物がもう直接見えない位置に移動した場合、それは地図上に表示され続ける。これは、— おそらく怪物が拾ったか腐ってしまったかして — 実際にはすでにそこになくても、あなたがその場所をもう一度見るか感じるかするまで残り続ける。注意すべき例外の一つは、その物が「覚えているけれども見えない怪物」マーカで覆われた場合である。あなたがそこに怪

物がいないことを確認してマークが消えると、見えない怪物が実際にその物を拾ったかどうかに関わらず、あなたはそこにあった物を忘れる。物がそこにあり続けている場合、再びその場所を見たり感知したりすると物を再び発見してそれを記憶し続ける。

この状況は物の山でも同様だが、山の一番上の物のみが表示される。山の上にいる敵に物を違う方法で表示するために、`hilite_pile` コマンドを有効に出来る。

## 8 制限プレイ

単に *NetHack* に勝利するだけでは満足できない一部のプレイヤーは、プレイに制限を設けることに挑戦している。これらの挑戦の一部はゲームによって自動的に記録され、ゲーム中いつでも `#conduct` コマンドで確認できる。またゲーム終了時にも表示される。挑戦に反するような行動を取ると、もはやその挑戦は表示されない。1 これらの挑戦に勝利することによって、プレイヤーは追加の“自慢する権利”を得られる。なお、これらの制限に従うことなくゲームに勝利することも完全に有効であり、最初にゲームに勝利するときは普通これらの挑戦は関係ない。

挑戦のいくつかは食べ物に関するものである。最も困難な挑戦は食料なしの挑戦である。生物は食料なしでも長い間生存できるが、水は必要である。従って、飲み物に対しては何の制限もない。例えば栄養分があってもである。神に祈って飢えをしのごことはいかなる食べ物に関する挑戦にも違反しない。

厳密な菜食主義者は動物から作られた食べ物を食べない。基本的な栄養源は果物と野菜である。ブロッコリー ('b')、ゼリー ('j')、細菌 ('F') も野菜とみなされる。人間の食料にも動物を使っていないものがある。レンバス、クラム、「食料」(丸薬)、Kレーション、Cレーションがそうである。他の生物に変化しているときは、金属などの普通消化できない物質も厳密な菜食主義者用の食料として扱われる。しかし、これらの物も食料なしの挑戦には反することに注意すること。

菜食主義者は動物を食べない。しかし、動物からの副産物を食べることは厳密な菜食主義者ほどは厳しくない。上記の厳密な菜食主義者が食べられるものに追加して、ブラックプリン以外のプリン ('P')、卵、卵から作られた食べ物 (古いクッキーとパンケーキ)、牛乳から造られた食べ物 (クリームパイとキャンディーバー)、ロイヤルゼリーを食べることができる。モンクは菜食主義者とみなされる。

肉を食べることは厳密な菜食主義者、菜食主義者、食料なしの挑戦に反する。これにはほし肉、上記以外の怪物の死体や缶詰、迷宮で見つかるその他の肉のかたまりを含む。変化中に「飲み込んで消化」攻撃をかけるのはその怪物の死体を食べたものとみなす。皮、竜鱗、骨でできた物体を、これらを消化できる怪物に変化して食べたり、マインドプレイヤーに変化して脳を食べることは動物を食べたとみなす。ただし、蜜蝋は動物の副生成物とみなす。

挑戦に関わらず、消化不可能な物もあり、食べると危険なものもある。怪物に対して「飲み込んで消化」攻撃をかけることは、その怪物の死体を食べたものとみなす。「厳格な菜食主義者 (“vegan”）」という言葉は食べ物に関する文脈でのみ用いられることに注意してほしい。動物から作られた物 (皮、竜の鱗、骨、角、さんご) を使ったり身につけたりすることは制限に違反しない。また、「ミルク色の」水葉は不透明な白色ではあるが、牛乳を含んでいるわけではないので、厳密な菜食主義者の制限に違反しない。スライムモルドやプレイヤーが定義した “fruits” は、例えば「さくらんぼ」であろうが「ポークチョップ」であろうが、厳密な菜食主義者の制限に違反しない。

無神論者は宗教を否定する。つまり、`#pray` (祈る)、`#offer` (いけにえを捧げる)、`#turn` (不死のものを戻す)、僧侶に対する `#chat` (話す) ことはできない。一部の読者はモンクや僧侶でプレイするのもこの制限に反していると主張するかもしれないが、ここではプレイヤーの選択の余地を残してある。イエンダーの魔除

けを神に捧げるのはゲームに勝利するために必要なので制限には反しないものとする。また、怒った神や僧侶やその他の宗教的存在が話す言葉を聞いても制限には反しない。真の無神論者は言葉は聞くが、そこになんら特別な意味を見出さない。

ほとんどのプレイヤーは武器 (あるいは武器として装備することを考慮している道具) を手に戦う。挑戦の一つはこのような武器を手にして使わずにゲームに勝利することである。武器を投げたり、発射したり、蹴ったりするのは許される。また、杖、呪文、あるいはその他のアイテムを使ったり、素手や脚で戦うことも許される。

*NetHack* において、平和主義者は他の怪物を殺してはならない (言い換えると、怪物の死によって経験値を得てはならない)。これは特に難しい挑戦であるが、他の手段によって経験値を得ることは可能である。

文盲者は読み書きができない。これには以下の行為が含まれる。巻き物を読む、魔法書を読む、占いクッキーのメッセージを読む、T シャツの文字を読む、巻き物を書く、“X” 一文字 (文盲者の伝統的なサイン) 以外の文字を刻む。刻んである文字を読むことと、ゲームに勝利するために絶対に必要なアイテムを読むことは制限には反しないものとする。ゲーム開始時に持っている巻き物と魔法書の内容を知っていることおよび呪文の知識は、ゲーム開始以前に教師から教わったものとみなし、制限には反しないものとする。

その他にいくつかの細かいゲームとして記録される挑戦がある。怪物は虐殺することが可能である。これをせずにプレイするのも挑戦とみなす。怪物を虐殺する機会が与えられたとき、“none” (“なし”) と答えると、虐殺しないままにすることができる。あなたはアイテムを同じグループの他のアイテムに変化させたり (“polypiling”)、変化の杖・呪文・薬で自分の体を他の生物に変えたり (“polyself”) できる。これらの効果を用いないことはそれぞれ挑戦とみなす。最後に、あなたはときどき願いをかなえてもらうことがある。なんのアイテムも願わずにプレイするのは一つの挑戦であり、聖器を願わない (たとえ直ちに消滅したとしても) でプレイするのも一つの挑戦である。アイテムを願う機会を与えられたとき、“nothing” (“なし”) と答えると、願わないままにすることができる。

## 9 オプション

人にはそれぞれいろいろな好みがあり *NetHack* の遊び方もそれぞれ異なっているので、*NetHack* の振る舞いを変更するため設定できるオプションがある。

### オプションの設定

オプションを設定する方法にはいくつかある。ゲーム中に ‘0’ コマンドを使うことによって全てのオプションを見ることができ、そのほとんどを変更できる。また、設定ファイルや環境変数 “NETHACKOPTIONS” で自動的に設定することもできる。*NetHack* のバージョンによってはゲーム開始前にオプションやシステム管理者のためのグローバルな設定をすることができるプログラムがついていることもある。

### 設定ファイルを使う

設定ファイルの標準設定のファイル名と位置は OS によって異なる。

Unix, Linux, Mac OS X では、これはユーザーのホームディレクトリの “.nethackrc” である。このファイルは存在していないかも知れないが、これは通常の ASCII テキストファイルで、任意のテキストエディ

タで作ることができる。

Windows では、“%USERPROFILE%\NetHack\3.6” フォルダの “.nethackrc” である。このファイルは存在していないかも知れないが、これは通常の ASCII テキストファイルで、任意のテキストエディタで作ることができる。NetHack の初回実行後、“%USERPROFILE%\NetHack\3.6” に “.nethackrc.template” というファイル名で、設定ファイルのデフォルトテンプレートを見つけれられるはずである。設定ファイルを作成していない場合、NetHack はデフォルトテンプレートファイルを使って設定ファイルを作成する。

MS-DOS では、nethack.exe と同じフォルダの “defaults.nh” である。

設定ファイルの中の ‘#’ で始まる行はコメントとして扱われる。空行は無視される。‘[’ で始まり ‘]’ で終わる行は節マークとして扱われる。大かっこで囲まれたテキストは節の名前である。節マーク以降の行はその節に属し、その節を選択するために CHOOSE 文が使われない限り無視される。節の名前は大文字小文字を無視する。

ファイル中に異なる設定文を使うことができ、そのいくつかは複数回使われる。一般的に、文は英大文字で書かれ、等号が続き、その文に特有の設定が続く。

以下は指定できる文の一覧である：

**OPTIONS** 真偽値オプションと複合オプションの 2 種類のオプションがある。真偽値オプションは設定のオンとオフを切り替える一方、複合オプションはより様々な値を取る。真偽値オプションに ‘no’ または ‘!’ を前置するとオフになる。複合オプションでは、オプション名と値はコロンで区切られる。一部のオプションは永続し、ゲーム開始時にのみ適用される。複数の OPTIONS 文を指定したり、一つの OPTIONS 文の中でカンマで区切ることで複数のオプションを指定したりできる。(カンマで区切られたオプションは右から左に処理される。)

例:

```
OPTIONS=dogname:Fido
OPTIONS=!legacy,autopickup,pickup_types:$"/!/?+
```

**HACKDIR** NetHack が必要なファイルのデフォルトの位置。Windows での HACKDIR のデフォルトは NetHack.exe または NetHackw.exe ファイルの位置なので、HACKDIR を上書きするように設定するのは通常不要で非推奨である。

**LEVELDIR** 進行中のレベルファイルを保管する位置。デフォルトは HACKDIR で、書き込み可能でなければならない。

**SAVEDIR** 保存されたゲームを保持する位置。デフォルトは HACKDIR で、書き込み可能でなければならない。

**BONESDIR** 骨ファイルを保持する位置。デフォルトは HACKDIR で、書き込み可能でなければならない。

**LOCKDIR** 同期ロックを保管する位置。デフォルトは HACKDIR で、書き込み可能でなければならない。

**TROUBLEDIR** ゲーム中段の記録とゲームの問題の自己診断を保持する位置。デフォルトは HACKDIR で、書き込み可能でなければならない。

- AUTOCOMplete** 拡張コマンドの自動補完をオンまたはオフにする。自動補完は X11 ウィンドウポートでは無効である。複数の自動補完を指定できる。自動補完をオンにするには、拡張コマンドの一覧を書く。コマンドに “!” を前置すると、そのコマンドの自動補完をオフにする。  
例:  

```
AUTOCOMplete=zap,!annotate
```
- AUTOPICKUP\_EXCEPTION** *pickup\_types* オプションの例外を設定する。「自動拾い例外の設定」の節を参照のこと。
- BINDINGS** 一部の特殊キー、メニューアクセラレータ、拡張コマンドのキー配置を変更する。複数の配置を指定できる。形式は、キーの後に、転んで区切られたコマンドが続く。さらなる情報については “Changing Key Bindings“ 節を参照のこと。  
例:  

```
BIND=~X:getpos.autodescribe
```
- CHOOSE** カンマで区切られたパラメータの一つを有効なセクション名としてランダムに選ぶ。他のセクションの行は無視される。  
例:  

```
OPTIONS=color
CHOOSE=char A,char B
[char A]
OPTIONS=role:arc,race:dwa,align:law,gender:fem
[char B]
OPTIONS=role:wiz,race:elf,align:cha,gender:mal
```
- MENUCOLOR** メニュー行を異なった色でハイライトする。“Configuring Menu Colors“ 節を参照のこと。
- MSGTYPE** 上部ステータス行に表示されるメッセージがどのように表示されるかを変更する。“Configuring Message Types“ 節を参照のこと。
- ROGUESYMBOLS** ログレベルのシンボル集合のためのカスタムシンボル。後述する *SYMBOLS* を参照のこと。
- SOUND** 効果音のマッピングを定義する。“Configuring User Sounds” 節を参照のこと。
- SOUNDDIR** 効果音ファイルがあるディレクトリを定義する。“Configuring User Sounds” 節を参照のこと。
- SYMBOLS** 特殊なログレベル以外の全てのダンジョンレベルで使われるシンボル集合のいくつかのシンボルを上書きする。“Modifying *NetHack* Symbols” 節を参照のこと。  
例:  

```
# replace small punctuation (tick marks) with digits
SYMBOLS=S_boulder:0,S_golem:7
```
- WIZKIT** デバッグモードのみ: 初期持物に追加される物。値は物の名前を 1 行一つ (最大 128 行) で書いたテキストファイルの名前である。各行は願いを扱う機能によって処理される。

例:

```
WIZKIT=~/.wizkit.txt
```

以下は設定ファイルの中身の例である:

```
# Set your character's role, race, gender, and alignment.
OPTIONS=role:Valkyrie, race:Human, gender:female, align:lawful

# Turn on autopickup, set automatically picked up object types
OPTIONS=autopickup,pickup_types:$="/!/?+

# Map customization
OPTIONS=color          # Display things in color if possible
OPTIONS=lit_corridor   # Show lit corridors differently
OPTIONS=hilite_pet,hilite_pile
# Replace small punctuation (tick marks) with digits
SYMBOLS=S_boulder:0,S_golem:7

# No startup splash screen. Windows GUI only.
OPTIONS=!splash_screen
```

## 環境変数 NETHACKOPTIONS を使う

環境変数 NETHACKOPTIONS にはいろいろなオプションの初期値をカンマで区切って列挙し設定する。オプションのうちのいくつかのものは単にオンかオフの選択ができるだけである。そのオプション名をリストに入れるとオンになり、オプション名の前に ‘!’ か “no” をつけるとオフになる。その他のオプションでは設定値として文字列が必要である。これらのオプションを設定するにはオプション名、コロンのまたはイコール記号、設定値の順で入力すればよい。設定値は次のカンマもしくは文字列の最後までとなる。

例えば *color* をオン、*legacy* をオフ、*name* が “Blue Meanie” に、*fruit* が “lime” になるように環境変数を設定するには *cs*h では次のコマンドを入力すればよい。(‘!’ は特殊文字なのでエスケープしてやる必要があることに注意)

```
setenv NETHACKOPTIONS "color,\!leg,name:Blue Meanie,fruit:lime"
```

*sh*, *ksh*, *bash* では

```
NETHACKOPTIONS="color,\!leg,name:Blue Meanie,fruit:lime"
export NETHACKOPTIONS
```

とすればよい。

NETHACKOPTIONS の値は事実上設定ファイル内の 1 行の OPTIONS 文と同じである。“OPTIONS=” 接頭辞が暗黙に含まれ、カンマ区切りのオプションは右から左に処理される。BIND や MSGTYPE のようなその他の設定文は使えない。

NETHACKOPTIONS は、カンマ区切りにオプションのリストの代わりに、使いたい設定ファイルのフルネームを設定できる。フルネームがスラッシュで始まっていない場合、‘@’ (アットマーク) を前置するこ

とで、残りがファイル名を意図していることを NetHack に伝える。‘/’ で始まっているなら、アットマークはオプションである。

## カスタマイズオプション

以下にいろいろなオプションの役割を説明する。あまりに長すぎる文字列の部分は無視される。以下のオプションのうち実装によっては無効となっているオプションもある。

一部のオプションは永続し、ゲーム中保存および再設定される。設定ファイルで永続オプションを変更した場合、新しいゲームにのみ適用される。

<i>acoustics</i>	あなたのキャラクターが聞いたものに関するメッセージを表示する (標準設定はオン)。これはコンピューターの音声出力機能とは何の関係もないことに注意すること。永続する。
<i>align</i>	スタート時の属性 ( <i>align:lawful</i> , <i>align:neutral</i> , <i>align:chaotic</i> )。最初の一字で指定することもできる。標準設定ではランダムに選択される。値に ‘!’ または “no” を前置すると、ランダムに選択した属性からその属性を除外できる。‘0’ コマンドで設定することはできない。永続する。
<i>autodescribe</i>	地図上の位置を指定するときにカーソルの下の地形を自動的に表現する (標準設定はオン)。 <i>whatis_coord</i> オプションは、表現に位置の座標を含むかどうかを制御する。
<i>autodig</i>	掘るための道具を手にしていて、掘ることができる地形に移動しようとしたとき、自動的に掘る (標準設定はオフ)。永続する。
<i>autoopen</i>	扉に向かって歩くとそれを開けようとする (標準設定はオン)。永続する。
<i>autopickup</i>	移動先にあるものを自動的に拾う (標準設定はオン)。永続する。この振る舞いをカスタマイズするためには “ <i>pickup_types</i> ” を参照すること。
<i>autoquiver</i>	このオプションは何も準備していない時に ‘f’ ( <i>fire</i> : 矢筒の中にあるものを発射する) コマンドを実行した時にどうするかを決める (標準設定はオフ)。オンなら、コンピューターは適当な武器を矢筒または矢筒袋に入れるか、適切な武器を準備する。この時、呪い、祝福、魔法、劣化、武器の数などは考慮されないので注意すること。代わりに ‘q’ コマンドを使って自分の好きな物を矢筒または矢筒袋に入れるか準備する。入れる武器がないか、このオプションがオフの時は代わりに ‘t’ ( <i>throw</i> : 物を投げる) コマンドが実行される。永続する。
<i>blind</i>	恒久的に盲目の状態ゲームを開始する (標準設定はオフ)。永続する。
<i>bones</i>	骨ファイルの読み書きを行う (標準設定はオン)。永続する。
<i>boulder</i>	巨岩を表示するのに用いるキャラクタ (標準設定は “large rock” クラス文字である ‘.’)。
<i>catname</i>	スタート時の猫の名前 (例: <i>catname:Morris</i> )。‘0’ コマンドで設定することはできない。
<i>character</i>	キャラクターのタイプを設定するための “ <i>role</i> ” の別名。 (例: “ <i>character:Monk</i> ”)。更なる詳細については <i>role</i> を参照のこと。
<i>checkpoint</i>	プログラムがクラッシュした時に復旧できるように階を移動する毎に状態をセーブする (標準設定はオン)。永続する。
<i>checkspace</i>	ファイルを書き込む前にディスクの空き容量をチェックする (標準設定はオン)。セーブ

ファイルやレベルファイルのために使うパーティションに 2GB 以上の空き容量がある場合、このオプションをオフをしなければならないかもしれない (なぜなら空き容量が大きすぎる場合、計算がオーバーフローして容量が足りないように見えるかもしれない)。このオプションはコンパイル時に MFLOPPY が定義されているときのみ適用される。

<i>clicklook</i>	マウスの右クリックで、画面上にあるものを見ることができるようにする (標準設定はオフ)。
<i>cmdassist</i>	ありがちなミスを検出したときに新規プレイヤーのために追加のコマンド補助メッセージを表示する (標準設定はオン)。
<i>confirm</i>	ペット、店主、その他の攻撃してこない生物を攻撃しようとしたときに確認をする (標準設定はオン)。永続する。
<i>dark_room</i>	明るい部屋の視界外のエリアを表示する (標準設定はオン)。永続する。
<i>disclose</i>	ゲーム終了時にプログラムがどの情報を公開するかを制御する。値は指示/カテゴリの組のスペース区切りのリストである (標準設定は 'ni na nv ng nc no' で、全ての候補に対してデフォルトを 'n' にする)。永続する。設定できるものは以下の通り。 i — "持ち物の公開" a — "属性の公開" v — "退治した怪物の一覧" g — "抹殺した怪物種の一覧" c — "挑戦の表示" o — "迷宮の概要の表示" それぞれの表示に対して、その前にどのように振舞うかを指示する文字をつけることができる。文字の意味は以下のとおり。 y — "確認するが、標準設定はイエスにする。" n — "確認するが、標準設定はノーにする。" + — "確認なしに表示する。" - — "確認なしで非表示とする。" 絶滅した怪物の一覧はソートできるので、'v' には二つの追加の選択肢がある: ? — "確認して、標準設定は問い合わせる。" # — "確認なしで表示し、ソート順を問い合わせる。" 問い合わせはメニューから順序の一つを選択する。確認なしの '+' の公開か、問い合わせに対して 'a' ではなく 'y' と答えた場合、伝統的な順序である、レベルの高いものから低いものの順で怪物を表示する。省略されたカテゴリには暗黙に 'n' 接頭辞が追加される。接頭辞が省略されて指定されたカテゴリには暗黙に '+' 接頭辞が使われる。表示カテゴリの順序は関係なく、プログラムは設定された順序に従ってゲーム終了時の表示を行う。(例: "disclose:yi na +v -g o") このように設定した場合、持ち物は確認する (標準設定はイエス)。属性は確認する (標準設定はノー)。退治した怪物は確認なしに表示する。抹殺した怪物は確認せずに非表示とする。挑戦は暗黙に確認する (標準設定はノー)。概要は確認せずに表示する。 退治した怪物の一覧は罾や同士討ちによって死んだ怪物も含むことに注意。そして迷宮の概要は、訪れた全てのレベルを表示するが、発見しなかったものについては開示され

	ない。
<i>dogname</i>	スタート時の犬の名前 (例: <code>dogname:Fang</code> )。‘0’ コマンドで設定することはできない。
<i>extmenu</i>	拡張コマンドインターフェースを、有効なコマンドの一覧メニューが出るように変更する。キー入力、最後に Enter を押す必要がないこと以外は伝統的なインターフェースと互換性がある。tty インターフェースに実装されている。(標準設定はオフ) 拡張コマンドを選ぶために常にメニューが表示される X11 インターフェースでは、これはメニューに利用可能な全てのコマンドを表示する (オン) か、伝統的に拡張コマンドと考えられているコマンドの部分集合のみを表示する (オフ) かを制御する。
<i>female</i>	“ <code>gender:female</code> ” の古い別名。‘0’ コマンドで設定することはできない。
<i>fixinv</i>	持ち物の目録記号と物との対応はその物を下に置いても変化しない (標準設定はオン)。オフにしたときは物を下に置くとその物の目録記号以降の記号と物との対応が 1 つずつずれる。永続する。
<i>force_invmenu</i>	持物を選ぶコマンドでは、可能なメニュー文字付きのテキスト問い合わせではなく、メニューを表示する。標準設定はオフ。
<i>fruit</i>	あなたの好物の果物にちなんでその名前を設定する (例: <code>fruit:mango</code> ) (標準設定は「 <code>slime mold</code> 」)。元来 <i>NetHack</i> で時々使われる懐古趣味的な妙な言葉である。slime mold(ネバネバかび) よりももっと食欲のわく食べ物にするべきであろう。apples(リンゴ)、oranges(オレンジ)、pears(洋なし)、bananas(バナナ)、melons (メロン) は <i>NetHack</i> にはもう存在しているので、設定してはならない。
<i>gender</i>	開始時の性別 ( <code>gender:male</code> または <code>gender:female</code> )。最初の一文字で指定することもできる。“male” と “female” オプションを指定していても、“gender” オプションが優先される。標準設定では適切な性別がランダムに選択される。値に ‘!’ または “no” を前置すると、ランダムに選択した性別からその性別を除外できる。‘0’ コマンドで設定することはできない。永続する。
<i>goldX</i>	物を祝祝状態 (BUCX) を基に絞り込むとき、金貨を X (祝祝状態不明、‘オン’ のとき)、あるいは U (呪われていないとわかっている、‘オフ’ のとき、標準設定) のどちらとして扱うかを設定する。金貨は決して祝福されたり呪われたりしないが、例えば <code>implicit_uncursed</code> オプションが ‘オフ’ でも “呪われていない” とは表示されない。
<i>help</i>	‘/’ コマンドを使って調べている物について何らかの情報がある場合、それを見るかどうかを尋ねる (標準設定はオン)。ヘルプが出ないようにすれば “More info?” というプロンプトに煩わされなくなるので、素早く物を調べることができる。しかしそれは興味深く重要な情報を見逃してしまうかも知れないことを意味する。永続する。
<i>herecmd_menu</i>	マウスに対応しているウィンドウポートを使っていて、自分自身か自分の隣をクリックしたとき、その位置に対して可能な公道のメニューを表示する。“ <code>#herecmdmenu</code> ” および “ <code>#therecmdmenu</code> ” コマンドと同じ。
<i>hilite_pet</i>	ペットを同種の動物から視覚的に区別する (標準設定はオフ)。このオプションの振る舞いは使っているウィンドウシステムに依存する。テキストウィンドウ表示では、ハイライトや反転がしばしば使われる。タイルを使用するときは、一般的にはハートの記号をペットにつける。curses インターフェースでは、 <code>petattr</code> オプションはどのようにペットをハイライトするかを制御し、これを設定すると <code>hilite_pet</code> オプションもこれに合わ

	せてオンオフされる。
<i>hilite_pile</i>	オブジェクトの山と個々のオブジェクトで表示を変える (標準設定はオフ)。このオプションの振る舞いは使用するウィンドウの種類に依存する。テキスト表示の場合、ハイライトや反転がしばしば使われる; タイルの場合、一般的には山の一番上のオブジェクトの横に小さいプラスマークが表示される。
<i>hitpointbar</i>	名前とランクの後ろにヒットポイントバーを表示する。TTY と Windows GUI 版のみで、ステータスハイライトがオンの場合のみ。
<i>horsename</i>	開始時の馬の名前をつける (例: “horsename:Trigger”)。 ‘0’ コマンドで設定することはできない。
<i>ignintr</i>	ブ레이크を含む割り込み信号を無視する (標準設定はオフ)。永続する。
<i>implicit_uncursed</i>	可能なら、持ち物一覧での「呪われていない」表示を抑制する (標準設定はオン)。
<i>legacy</i>	ゲーム開始時に説明メッセージを表示する (標準設定はオン)。永続する。
<i>lit_corridor</i>	夜目やあなたのキャラクターが持っている光源によって見える通路を表示する (標準設定はオフ)。永続する。
<i>lootabc</i>	箱を扱うときに ‘o’, ‘i’, ‘b’ ではなく古い ‘a’, ‘b’, ‘c’ のキーを使う (標準設定はオフ)。永続する。
<i>mail</i>	ゲーム中にメールを配達するようにする (標準設定はオン)。永続する。
<i>male</i>	“gender:male” の古くなった別名。 ‘0’ コマンドで設定することはできない。
<i>mention_walls</i>	壁に向かって歩くときにフィードバックを与える (標準設定はオフ)。
<i>menucolors</i>	メニュー行に色を付ける (標準設定はオフ)。色の設定方法については “ <i>Configuring Menu Colors</i> ” を参照のこと。
<i>menustyle</i>	いろいろなオブジェクトを指定するとき (たとえば Drop コマンド) に用いるインタフェースの制御。値は traditional, combination, full, partial の 4 つのタイプの最初の一字を指定する。Traditional は古いバージョンで唯一利用できたもので、物の種類を示す文字の入力、次に選んだ種類に当てはまる全ての物一つずつ確認する。Combination は選択したい物の種類を示す文字を入力するが、次に当てはまる物一つずつ確認で選択するのではなく、メニュー形式で選択する。Full は物の種類の文字入力かわりに、最初に物の種類のメニューを表示し、それから選択した種類に当てはまる物のメニューを表示する。Partial は物の種類を選ぶことをせず、直ちに全ての物が表示されたメニューを表示する。永続する。
<i>menu_deselect_all</i>	メニューの全ての項目を非選択にするキー。Amiga, Gem, X11, tty で実装されている。(標準設定は ‘-’)
<i>menu_deselect_page</i>	メニューのうち、現在表示されている全ての項目を非選択にするキー。Amiga, Gem, tty で実装されている。(標準設定は ‘\’)
<i>menu_first_page</i>	メニューの最初のページへ移動するキー。Amiga, Gem, tty で実装されている。(標準設定は ‘^’)
<i>menu_headings</i>	メニューのうち注目している部分をどのように表示するかを指定する。値は “none”, “bold”, “dim”, “underline”, “blink”, “inverse” のいずれかである。全てのポート

で実際に全てのタイプを表示できるわけではない。

<i>menu_invert_all</i>	メニューの全ての項目の選択状態を反転するキー。Amiga, Gem, X11, tty で実装されている。(標準設定は '@')
<i>menu_invert_page</i>	メニューの現在のページの全ての項目の選択状態を反転するキー。Amiga, Gem, tty で実装されている。(標準設定は '~')
<i>menu_last_page</i>	メニューの最後のページへ移動するキー。Amiga, Gem, tty で実装されている。(標準設定は ' ')
<i>menu_next_page</i>	メニューの次のページへ移動するキー。Amiga, Gem, tty で実装されている。(標準設定は '>')
<i>menu_objsyms</i>	物の種類の記号がメニューのショートカットキーとして動作する場所では、メニューの一覧に物の種類の記号を表示する (標準設定はオフ)。
<i>menu_overlay</i>	メニューを書く前に画面をクリアせず、メニューを画面の右端に寄せる。tty 版のみ。(標準設定はオン)
<i>menu_previous_page</i>	メニューの前のページへ移動するキー。Amiga, Gem, tty で実装されている。(標準設定は '<')
<i>menu_search</i>	メニューで検索を行うキー。Amiga, Gem, X11, tty で実装されている。(標準設定は ':')
<i>menu_select_all</i>	メニューの全ての項目を選択にするキー。Amiga, Gem, X11, tty で実装されている。(標準設定は ':')
<i>menu_select_page</i>	メニューのうち、現在表示されている全ての項目を選択にするキー。Amiga, Gem, tty で実装されている。(標準設定は ';')
<i>monopolycontrol</i>	怪物が姿を変えるときに新しい形になるか確認する (標準設定はオフ)。デバッグモードのみ。
<i>mouse_support</i>	入力と移動にマウスを使えるようにする。有効な設定は: 0 — 無効 1 — 有効; マウスの使用を支援するために OS の補正を使う 2 — 1 と同様だが、OS の補正は使わない
	値を省略すると 1 を指定したのと同じになり、 <i>mouse_support</i> を否定すると 0 を指定したのと同じになる。
<i>msghistory</i>	後で '^P' で呼び出せるように保存しておく最上行のメッセージの数 (標準設定は 20)。 '0' コマンドで設定することはできない。
<i>msg_window</i>	過去のメッセージの表示方法を変更する。現時点では tty (四つ全ての選択肢) と curses ('f' と 'r' の選択肢、標準設定は 'r') で実装されている。指定できる値は以下のとおり。 s — "single": 1 メッセージだけ (標準設定。これは 3.4.0 以前での振る舞いである) c — "combination": 最初の 2 メッセージは 'single'、その後は 'full'。 f — "full": 全画面; 古いメッセージが先。 r — 全画面; 新しいメッセージが先。 過去との互換性のために、値なしにもできる ( <i>full</i> とみなす) し、否定もできる ( <i>single</i>

とみなす)。

<i>name</i>	あなたのキャラクターの名前を設定する (標準設定はあなたの ユーザー名)。-記号とキャラクターの職業の頭文字を付け加える (つまり “-A -B -C -H -K -M -P -Ra -Ro -S -T -V -W” のどれかを後に付ける) ことによって職業を設定することもできる。職業として “-@” を使用すると、ランダムなひとつが自動的に選択される。‘0’ コマンドで設定することはできない。
<i>news</i>	<i>NetHack</i> のニュースファイルがあればそれを読む (標準設定はオン)。ニュースはゲームの最初に表示されるので、‘0’ コマンドでこれを設定するのは無意味である。
<i>nudist</i>	防具なしのキャラクターで開始する (デフォルトはオフ)。永続する。
<i>null</i>	端末にパディングのためのナルキャラクターを送る (標準設定はオン)。永続する。
<i>number_pad</i>	移動のために文字のかわりに数字キーを使用する (標準設定は 0 またはオフ)。設定可能な値は: 0 — 文字で移動する; ‘yuhjklbn’ 1 — 数字で移動する; 数字 ‘5’ は ‘G’ 移動の接頭辞として振る舞う 2 — 1 と同様だが数字 ‘5’ は ‘G’ ではなく ‘g’ 移動の接頭辞として振る舞う 3 — 電話のキー配置の数字で移動する; 123 は上、789 は下 4 — 3 と 2 の組み合わせ; 電話配置と MS-DOS 互換 -1 — 文字で移動するが ‘z’ で北西に移動し、‘y’ で杖を振る 後方互換性のために、値を省略するのは 1 と同じで、否定する <i>number_pad</i> と 0 を指定するのと同じである。(設定 2 と 4 は MS-DOS や古い PC Hack との互換性のためのものである; ‘5’ のための異なった振る舞いに加えて、‘Alt-5’ は ‘G’ として振る舞い、‘Alt-0’ は ‘I’ として振る舞う。設定 -1 は、‘y’ と ‘z’ の位置が入れ替わっている一部の QWERTZ キーボードに対応するものである。) 数字で移動するとき、(12 回調べるために “12s” とするような) コマンドに対する回数接頭辞を入力するには、文字 ‘n’ を前置する (“n12s”)。
<i>packorder</i>	物の種類を表示するときの順番を指定する (標準設定は “”) [%?+!=(*‘0_”)。このオプションにはいろいろな物の種類を表す文字の列を設定する。設定されなかった種類は以前の順序で最後に表示される。
<i>paranoid_confirmation</i>	プロンプトを変えたい状況の、スペースで区切られた一覧。標準設定は “ <i>paranoid_confirmation:pray</i> ”。
	Confirm ‘y’ ではなく “yes” を必要とする場合、yes 以外の何でも否定したことにするのではなく、否定する場合には “no” が必要。
	quit ゲームを抜けたリスコアが記録されない探検モードに切り替えるときに ‘y’ ではなく “yes” が必要;
	die 死ぬときに ‘y’ ではなく “yes” が必要 (通常のプレイでは無意味; 探検モードで適用される);
	bones デバッグモードで死んだときの骨ファイルの保存確認に ‘y’ ではなく “yes” が必要;
	attack 友好的な怪物を攻撃するときの確認に ‘y’ ではなく “yes” が必要;
	wand-break 杖を折るときの確認に ‘y’ ではなく “yes” が必要;

eating 食べ続けるかどうかの確認に 'y' ではなく "yes" が必要;  
 Were-change プレイヤーが変化制御を持っているときに獣化病により姿を変えるときの確認に 'y' ではなく "yes" が必要;  
 pray 祈るときにすぐに祈るのではなく確認で 'y' が必要; 標準設定でオン;  
 Remove 'R' と 'T' コマンドで有効なアイテムを一つしか身につけていないときでも持ち物一覧からの選択が必要。  
 all 全て有効にする。

標準設定では、pray 選択がオン、それ以外はオフになっている。その他の選択を設定することなくこれをオフにするには、"*paranoid\_confirmation:none*" を使う。その他を設定しながらこれをオンのままにするには、"*paranoid\_confirmation:attack pray Remove*" のように一覧に含める。

*perm\_invent* もしオンならば、現在の持ち物一覧を常にウィンドウに表示しておく。この機能が実装されているウィンドウシステムでのみ有効。

*petattr* 地図上にペットを表示するときに使われるハイライト属性の一つ以上のテキストを指定する。事実上 *hilite\_pet* オンオフオプションの上位互換である。curses インターフェースのみ; 値は一つ以上の次の文字である。

n — 通常のテキスト (ハイライトなし)  
 i — 反転 (標準設定)  
 b — 太字  
 u — 下線  
 k — 点滅  
 d — 薄い表示  
 t — 斜体  
 l — 左の行指示子  
 r — 右の行指示子

ターミナルハードウェアやターミナルエミュレーションソフトウェアに依存して、これらの選択の一部、特に最後の三つは動作しないかもしれない。

現在の所、複数のハイライトスタイル文字は単に結合されている (例えば "bk") が、将来プラス記号で区切られる必要があるようになるかもしれない (例えば "b+k" のように; これは既に動作する)。*'n'* の選択を使う場合、その他の文字との組み合わせではなく、単体で指定される必要がある。

*pettype* プレイするキャラクタクラスが複数のタイプのペットを使用できる時に初期のペットを指定する、または初期のペットを全く指定しない。指定出来るのは "cat", "dog", "horse", "none" である。選択が現在プレイしている職業で使えない場合、暗黙に無視される。例えば、"horse" は騎士をプレイしている場合にのみ有効になる。*'0'* コマンドで設定することはできない。

*pickup\_burden* 物を拾った時にここで指定した荷物の重さレベル (Unencumbered, Burdened, streSsed, straiNed, overTaxed, overLoaded) 以上になると、続行するかどうかを確認する。(標準設定は 'S')。永続する。

*pickup\_thrown* このオプションがオンで "*autopickup*" もオンの場合、例え "*pickup\_types*" にないか自

	動拾い例外にマッチングしても、自分が投げた物は拾おうとする。デフォルトはオン。永続する。
<i>pickup_types</i>	“ <i>autopickup</i> ” オプションが設定されている時に拾う物の種類を指定する (標準設定は全ての種類)。“ <i>autopickup_exception</i> ” 行を使うことで“ <i>autopickup</i> ” の振る舞いをより細かく制御できる。永続する。
<i>pile_limit</i>	床の上の物の山の上を歩くとき、物の一覧を表示する代わりに “there are few/several/many objects here” (ここにはいくつかの/たくさんの物がある) を表示する閾値。値 0 は「無制限」(常に物の一覧を表示する) を意味する; 値 1 は、山のサイズが最低 1 はあるの、事実上「物は一切表示しない」を意味する; 標準設定は 5。永続する。
<i>playmode</i>	値は <i>normal</i> , <i>explore</i> , <i>debug</i> のいずれか。通常のプレイの代わりに (発見モードとも呼ばれる) 探検モード、あるいは (ウィザードモードとも呼ばれる) デバッグモードを選択する。デバッグモードは、(マルチユーザーシステムでは) 特定のユーザー名でログインしている必要があるかもしれないし、(シングルユーザーシステムでは) 特定のキャラクタ名を指定している必要があるかもしれないし、完全に無効になっているかもしれない。デバッグモードを要求したけれども許されていないか不可能な場合は代わりに探検モードになる。標準設定は通常プレイ。
<i>pushweapon</i>	既に武器を持っている時に ‘w’ (wield: 武器を持つ) コマンドを使ったとき、既に持っていた武器を予備の武器に設定する (標準設定はオフ)。アイテムを使うことで装備したことになる場合は ‘a’ (apply) コマンドと同様である。永続する。
<i>race</i>	種族を選択する (例: “ <i>race:human</i> ”)。標準設定はランダムである。値に ‘!’ または “no” を前置すると、ランダムに選択した種族からその種族を除外できる。‘0’ コマンドで設定することはできない。永続する。
<i>rest_on_space</i>	スペースキーを ‘.’(#wait) コマンドとして使用する (標準設定はオフ)。永続する。
<i>role</i>	キャラクターのタイプを設定する (例: “ <i>role:Samurai</i> ”)。“ <i>character</i> ” オプションと同じである。職業を設定するその他の方法については “ <i>name</i> ” オプションを参照のこと。普通は先頭の 1 文字だけで判別されるが、‘r’ は例外である。“ <i>Rogue</i> ”、“ <i>Ranger</i> ”、“ <i>random</i> ” があるからである。値に ‘!’ または “no” を前置すると、ランダムに選択した職業からその職業を除外できる。‘0’ コマンドで設定することはできない。永続する。
<i>roguesymset</i>	このオプションは、rogue レベルで表示されるシンボルを変更するために、 <i>symbols</i> の中にある名前付きシンボル集合の一つを選択するために使われる。
<i>rlcomp</i>	セーブファイルを書き出すときに、地図にランレングス圧縮を行う。ランレングス圧縮に対応していないシステムもある。既存のセーブファイルを読み込むときには影響しない。
<i>runmode</i>	「まとめて移動」する (shift+方向や control+方向で移動するか、旅行コマンドやマウスのクリックを使った場合) ときにどれくらいの頻度で地図を更新するかを制御する。設定可能な値は以下の通り: <i>teleport</i> — 移動が完了してから地図を更新する; <i>run</i> — 7 歩ぐらい毎に地図を更新する; <i>walk</i> — 1 歩毎に地図を更新する; <i>crawl</i> — <i>walk</i> と同様だが、一歩毎にしばらく停止する。

	このオプションは画面表示にのみ影響し、実際の移動結果には影響しない。標準設定は <i>run</i> である。バージョン 3.4.1 以前は <i>teleport</i> のみであった。効果が確認できるかどうかは使っている版や端末の種類に依存する。永続する。
<i>safe_pet</i>	ペットを (ペットと知りつつ) 攻撃してしまうのを防ぐ (標準設定はオン)。永続する。
<i>sanity_check</i>	各ターンの前に怪物、物、地図を評価する (標準設定はオフ)。デバッグモードのみ。
<i>scores</i>	最後にスコアリストのどの部分を見るかを制御する (例: “ <i>scores:5top scores/4around my score/own scores</i> ”)。それぞれの分野の最初の文字 (‘t’, ‘a’, ‘o’) のみが必要である。永続する。
<i>showexp</i>	最下行に現在の経験点を表示する (標準設定はオフ)。永続する。
<i>showrace</i>	あなたを表示するときに、職業に対応するマークではなく、種族に対応するマークで表示する (標準設定はオフ)。この設定は表示にだけ用いられ、ゲームとしては何も変わらないことに注意。永続する。
<i>showscore</i>	最下行に現在のスコアを表示する (標準設定はオフ)。永続する。
<i>silent</i>	端末のピープ音を鳴らさない (標準設定はオン)。永続する。
<i>sortloot</i>	持ち物一覧や <i>#loot</i> コマンドやその他いくつかの場合でのソートの振る舞いを制御する。永続する。 可能な値は: <i>full</i> — 常に一覧をソートする; <i>loot</i> — <i>#loot</i> コマンドや広くコマンドのように、持ち物を示す文字を使わないときにだけソートする; <i>none</i> — ソートなしの伝統的な方法で一覧を表示する。
<i>sortpack</i>	持ち物の目録を表示するとき種類毎に荷物の内容を並べ替える (標準設定はオン)。永続する。
<i>sparkle</i>	怪物 (やあなた) が攻撃を受け、それに抵抗した場合に画面効果を表示する (標準設定はオン)。永続する。
<i>standout</i>	怪物と “--More--” を太字で表示する (標準設定はオフ)。永続する。
<i>statahilites</i>	何ターンステータスハイライトでフィールドをハイライトするかを制御する。負またはゼロを指定すると、ステータスハイライトを無効にする。さらなる情報については “ <i>Configuring Status Hilites</i> ” を参照のこと。
<i>status_updates</i>	画面の下部でのステータス行の更新を認める (標準設定はオン)。
<i>suppress_alert</i>	前のバージョンから変更された機能に対して注意を促すメッセージを表示しないようにする <i>NetHack</i> のバージョンを設定する。 (例: “ <i>suppress_alert:3.3.1</i> ”) 例のように指定すると 3.3.1 以前に変更された機能に対する注意メッセージは表示されない。
<i>symset</i>	このオプションは、画面に表示される文字を変更するために、 <i>symbols</i> にある名前付きシンボル集合の一つを選択するために使われる。標準の文字を選択するには明示的に “ <i>symset:default</i> ” を使うこと。
<i>time</i>	最下行にゲームで経過した時間をターン数で表示する (標準設定はオフ)。永続する。
<i>timed_delay</i>	爆発や物体の移動など、効果を表示するためにちょっと時間待ちするとき、特殊なキャ

	ラクタを画面に送るかわりにタイマーを使用する。(“tty” インタフェースにのみ適用される。“X11” インタフェースは常にタイマーを使用する。標準設定はタイマーによる時間待ちが組み込まれているならばオン) 永続する。
<i>tombstone</i>	死んだとき墓石を表示する (標準設定はオン)。永続する。
<i>toptenwin</i>	ゲーム終了時の表示を標準出力のかわりに <i>NetHack</i> のウィンドウに表示する (標準設定はオフ)。このオプションを設定するとウィンドウを使うバージョンの <i>NetHack</i> では起動した親ウィンドウとは別のウィンドウにスコアリストが表示される。しかし、スコアリストはゲームが終了したあとターミナルやウィンドウには残らない。
<i>travel</i>	トラベルコマンドを有効にする (標準設定はオン)。このオプションをオフにすることによって、うっかり地図ウィンドウをクリックすることによって、望まない移動をしようとするのを防ぐことができる。永続する。
<i>verbose</i>	ゲーム中のコメントを詳細に表示する (標準設定はオン)。永続する。
<i>whatis_coord</i>	When using the “autodescribe” がオンのときに地図上を見て回るために ‘/’ や ‘;’ コマンドを使うとき、説明の後ろに座標を表示する。場所を選択するその他の状況でも動作する。
	<p>可能な設定は:</p> <p>c — コンパス (‘east’, ‘3s’, ‘2n,4w’);</p> <p>f — 完全コンパス (‘east’, ‘3south’, ‘2north,4west’);</p> <p>m — 地図 &lt;x,y&gt; (地図列 x=0 は使われない);</p> <p>s — 画面 [row,column] (row は tty 利用法に一致するオフセット);</p> <p>n — なし (座標は表示されない) [標準設定].</p> <p><i>whatis_coord</i> オプションは、‘/’ の ‘/m’, ‘/M’, ‘/o’, ‘/O’ サブコマンドでも使用できる; ここで ‘none’ 設定は ‘map’ で上書きされる。</p>
<i>whatis_filter</i>	<p>地図上の場所を指定するときに、目標の中を循環して移動するキーのいずれかを使っているとき、可能性のある目標をフィルタリングする。(標準設定はなし)</p> <p>可能な設定は:</p> <p>n — フィルタリングなし;</p> <p>v — 視界内のみ;</p> <p>a — 同じエリア (部屋、通路など). エリアフィルタはやや予言的である – 出入り口に立っている場合、移動してきたエリア側にいるものとして扱われる。フィルタリングは、“getpos.filter” キーで場所を指定するときにも変更できる。</p>
<i>whatis_menu</i>	地図上の場所を指定するときに、目標の中を循環して移動するキーのいずれかを使っているとき、代わりにメニューを表示する。(標準設定はオフ)
<i>whatis_moveskip</i>	地図上の場所を指定するときに、高速移動のためにシフトを押しながらの移動や、meta-数字 キーでの移動を使っているとき、8 マス毎ではなく同じ記号を飛ばして移動する。(標準設定はオフ)
<i>windowtype</i>	プログラムが複数のインターフェースに対応してビルドされているとき、ウィンドウシステムを使用するかどうかを「tty」または「X11」で選択する (標準設定はビルド時の設定による; “#version” でチェックできる)。「0」コマンドで設定することはできない。

使われる場合は、これは設定される最初のオプションである必要がある; この値はその他の様々なオプションの可用性を有効にしたり無効にしたりするかもしれないからである。設定ファイルに複数行ある場合、これは最初の非コメント行である。NETHACKOPTIONS や設定ファイル中の OPTIONS 行でのカンマ区切りリストについては、リストの最も右のオプションである。

*wizweight* 物の説明にその物の重さを追加する (標準設定はオフ)。デバッグモードのみ。  
*zerocomp* セーブファイルを書くときに、内容に zero-comp 圧縮を掛ける。全てのポートで zero-comp 圧縮に対応しているわけではない。既存のセーブファイルを読み込むときには効果はない。

## ウィンドウシステム版専用カスタマイズオプション

以下はあなたが選択したウィンドウタイプの特徴をカスタマイズして変更するために用いる様々なオプションの説明である。長すぎる文字列は適当に切り詰められる。全てのウィンドウタイプに対してこれら全てのオプションが有効というわけではない。これらのオプションが設定されたとき、もしウィンドウタイプがその設定を受け付けるなら、そうされる。もし受け付けられないなら、単に無視される。現在のウィンドウタイプで対応しているオプションは、オプション一覧表示 ('0') で表示される。一部のオプションは '0' コマンドで動的に変更できる。

*align\_message* メッセージウィンドウをどこに置くか (top, bottom, left, right)  
*align\_status* ステータスウィンドウをどこに置くか (top, bottom, left, right)  
*ascii\_map* 可能なら ASCII キャラクタマップを表示する  
*color* 可能なら怪物、物体、洞窟の構成要素をカラーで表示する。  
*eight\_bit\_tty* 可能なら 8 ビットキャラクタ (例えば *traps* オプションで指定したもの) をそのままターミナルに表示する (標準設定はオフ)。  
*font\_map* 可能ならマップウィンドウにここで選んだ名前のフォントを使う。  
*font\_menu* 可能ならメニューウィンドウにここで選んだ名前のフォントを使う。  
*font\_message* 可能ならメッセージウィンドウにここで選んだ名前のフォントを使う。  
*font\_status* 可能ならステータスウィンドウにここで選んだ名前のフォントを使う。  
*font\_text* 可能ならテキストウィンドウにここで選んだ名前のフォントを使う。  
*font\_size\_map* 可能ならマップウィンドウにこのフォントサイズを使う。  
*font\_size\_menu* 可能ならメニューウィンドウにこのフォントサイズを使う。  
*font\_size\_message* 可能ならメッセージウィンドウにこのフォントサイズを使う。  
*font\_size\_status* 可能ならステータスウィンドウにこのフォントサイズを使う。  
*font\_size\_text* 可能ならテキストウィンドウにこのフォントサイズを使う。  
*fullscreen* 可能ならウィンドウでなく画面全体で表示しようと試みる。  
*guicolor* (メニュー選択の文字のような) 一部の非マップデータを表示するときにテキストの色やハイライト属性を使う。curses インターフェースのみ; 標準設定はオン。  
*large\_font* 可能なら大きいフォントを使う。

<i>map_mode</i>	可能なら指定された方法でマップを表示する。
<i>player_selection</i>	可能ならキャラクター選択時にダイアログボックスや確認画面を用いる。
<i>popup_dialog</i>	可能なら入力時にポップアップダイアログボックスを用いる
<i>preload_tiles</i>	可能ならタイルをメモリに予め読み込んでおく。例えば、プロテクトモード MS-DOS 版の場合、ゲーム開始時にタイルを RAM に予め読み込むかどうかを制御する。予め読み込むとタイルグラフィックの性能は向上するが、より多くのメモリを消費する (標準設定はオン)。「0」コマンドで設定することはできない。
<i>scroll_amount</i>	可能なら、 <i>scroll_margin</i> オプションで指定された位置に来たときに、どれだけの数スクロールさせるかを指定する。
<i>scroll_margin</i>	可能なら、ウィンドウの端からここで指定されたマス数にあなたまたはカーソルが移動したときに、画面をスクロールさせる。
<i>selectsaved</i>	可能なら、ゲーム開始時に、可能ならプレイヤーに選択させるために既存のセーブしたゲームのメニューを表示する。全てのポートがこのオプションに対応しているわけではない。
<i>softkeyboard</i>	可能なら、ソフトウェアキーボードを表示する。ハンドヘルドはおそらくこのオプションに対応しているであろう。
<i>splash_screen</i>	可能なら、起動時にスプラッシュスクリーンを表示する (標準設定はオン)
<i>statuslines</i>	伝統的な地図の下のステータス表示の行数。設定可能な値は 2 と 3 (標準設定は 2)。 <i>curses</i> と <i>tty</i> インターフェースのみ。
<i>term_cols</i>	and
<i>term_rows</i>	<i>curses</i> インターフェースのみ。表示に使う桁数と行数。 <i>curses</i> は指定された値にリサイズしようとするが、大きすぎる場合にはより小さいサイズに合わせる。標準設定は現在のウィンドウサイズ。
<i>tiled_map</i>	可能なら、タイルマップを表示する。
<i>tile_file</i>	標準設定を上書きする別のタイルファイルの名前
<i>tile_height</i>	タイルが表示できる環境でのタイルの高さ
<i>tile_width</i>	タイルが表示できる環境でのタイルの幅
<i>use_darkgray</i>	黒記号のために青ではなくボールドの黒を使う (TTY のみ)。
<i>use_inverse</i>	可能なら、ゲームが指定したときに画面を反転する。
<i>vary_msgcount</i>	可能なら、メッセージウィンドウに一度に表示するメッセージの数を指定する。
<i>windowborders</i>	マップ、ステータスエリア、メッセージエリア、および有効の場合は固定持ち物ウィンドウの周りに箱を書くかどうか。 <i>curses</i> インターフェースのみ。可能な値は 0 — オフ、常に境界を表示しない 1 — オン、常に境界を表示する 2 — 自動、最低 $(24+2) \times (80+2)$ ある場合に表示 (標準設定)

(‘2’ の  $26 \times 82$  というサイズの閾値は、画面の行数と列数から決定されている。*align`status* を *left* または *right* に設定するには、最低 110 桁  $(80+2+26+2)$  の幅が必要である。



<i>subkeyvalue</i>	(Win32 tty NetHack のみ)。国際キーボードの問題を補正するのを助けるために、OS が NetHack に返すキー入力の変更に使用する。OPTIONS=subkeyvalue:171/92 とすると、もともと 171 が返されようとしていた場合、NetHack に 92 を返す。必要なら、複数の subkeyvalue 行を設定ファイルに書いてもよい。‘0’ コマンドで設定することはできない。
<i>video</i>	使用するビデオモードを設定する (PC NetHack のみ)。値は autodetect, default, vga のいずれかである。vga (または VGA ハードウェアがあるときに autodetect) に設定すると、表示にタイルを用いる。‘0’ コマンドで設定することはできない。
<i>videocolors</i>	NO_TERMS を使用している PC システムのカラーパレットをセットする (標準設定は「4-2-6-1-5-3-15-12-10-14-9-13-11」PC NetHack のみ)。シンボルの順番は、赤・緑・茶色・青・マゼンタ・シアン・輝く白・輝く赤・輝く緑・黄色・輝く青・輝くマゼンタ・輝くシアン。‘0’ コマンドで設定することはできない。
<i>videoshades</i>	3 段階の利用可能なグレイスケールを設定する (標準設定は dark normal light; PC NetHack のみ)。もしゲーム画面が見にくい場合は、3つのスケールを調整してみる。もしこれでうまくいかなかったら、!color を試してみる。‘0’ コマンドで設定することはできない。

## 正規表現

正規表現は通常 POSIX 拡張正規表現である。正規表現ライブラリがないプラットフォームでは正規表現対応なしで NetHack をコンパイルすることも可能である。これは最近のプラットフォームでは真ではないが、この方法で NetHack がビルドされると、パターンは代わりにグロブパターンが使われる。これは自動拾い例外、メッセージ型、メニュー色、ユーザー音声に適用される。

## 自動拾い例外の設定

“autopickup” オプションの振る舞いを “pickup\_types” オプションを使ってより洗練させることができる。

“autopickup\_exception” 行を設定ファイルに書くことで、何かを拾おうとするときにチェックすべきパターンを定義できる。

*autopickup\_exception* ‘pickup\_types’ オプションの例外を設定する。autopickup\_exception オプションはあなたの位置にある物体の説明の単数形に一致するパターンを指定する。これは正規表現であるべきである。

さらに、パターンの先頭に置いた場合に特別扱いされる文字もある。

< — 以下のパターンに一致する物体は常に拾う。

> — 以下のパターンに一致する物体は決して拾わない。

autopickup\_exception 規則は、設定ファイルで現れた順に処理されるので、後の規則で前の規則を上書きできる。

‘0’ コマンドで設定することもできるが、これらは設定ファイルに含まれないので、ゲームをセーブして復元すると適用されない。autopickup\_exception ルールはゲームと共に保存されない。

以下は例である:

```
autopickup_exception="<*arrow"  
autopickup_exception=">*corpse"  
autopickup_exception=">*cursed*"
```

上記の例の最初のは、全ての矢を自動的に拾う。二番目の例は、死体は自動拾いの例外とする。最後の例は、呪われているとわかっているものは自動拾いの例外とする。

## キー配置を変更する

設定ファイルで BIND 文を使うことにより、一部の特殊コマンド、メニューアクセラレータキー、拡張コマンドのデフォルトのキーバインディングを変更できる。形式は、キーの後に、コロンの区切られたバインド先のコマンドが続く。キーには、単一文字 (「x」)、制御キー (“^X”, “C-x”)、メタキー (“M-x”)、または 3 桁の 10 進 ASCII コードを使用できる。

例:

```
BIND=^X:getpos.autodescribe  
BIND={:menu_first_page  
BIND=v:loot
```

**拡張コマンドキー** 複数のキーを一つの拡張コマンドに割り当てることができる。割り当てられている拡張コマンドとして “nothing” を使うことでキーへの割り当てを解除する。また、“<esc>”, “<enter>”, “<space>” キーにも割り当てることができる。

**メニューアクセラレータキー** メニュー制御やアクセラレータキーも設定ファイルの OPTIONS 行を通して再割り当てできる。物のシンボルをメニューアクセラレータに割り当ててはできない。

**特殊コマンドキー** 以下は再割り当てできる特殊コマンドである。その一部は問題なく同じキーに割り当てることができる; それ以外は同じ「コンテキスト」にあり、同じキーに割り当てると、そのうち一つだけが利用可能である。特殊コマンドは単一のキーにのみ割り当てることができる。

- count** あるコマンドを複数回繰り返すための回数を開始するための前置キー。  
*number\_pad* がオンの場合のみ。標準設定は ‘n’。
- doinv** 持物を表示する。*number\_pad* がオンの場合のみ。標準設定は ‘0’。
- fight** 強制的にある方向に戦うための前置キー。標準設定は ‘F’。
- fight.numpad** 強制的にある方向に戦うための前置キー。*number\_pad* がオンの場合のみ。標準設定は ‘-’。
- getdir.help** 方向を訊ねられたとき、ヘルプを表示するキー。標準設定は ‘?’。
- getdir.self** 方向を訊ねられたとき、自分自身に向けるキー。標準設定は ‘.’。
- getdir.self2** 方向を訊ねられたとき、自分自身に向けるキー。標準設定は ‘s’。
- getpos.autodescribe** 場所を訊ねられたとき、*autodescribe* を切り替えるキー。  
標準設定は ‘#’。

**getpos.all.next** 場所を訊ねられたとき、次に近い興味のあるものに移動するキー。標準設定は 'a'。

**getpos.all.prev** 場所を訊ねられたとき、前に近い興味のあるものに移動するキー。標準設定は 'A'。

**getpos.door.next** 場所を訊ねられたとき、次に近いドアや出入り口に移動するキー。標準設定は 'd'。

**getpos.door.prev** 場所を訊ねられたとき、前に近いドアや出入り口に移動するキー。標準設定は 'D'。

**getpos.help** 場所を訊ねられたとき、ヘルプを表示するキー。標準設定は '?'。

**getpos.mon.next** 場所を訊ねられたとき、次に近い怪物に移動するキー。標準設定は 'm'。

**getpos.mon.prev** 場所を訊ねられたとき、前に近い怪物に移動するキー。標準設定は 'M'。

**getpos.obj.next** 場所を訊ねられたとき、次に近い物に移動するキー。標準設定は 'o'。

**getpos.obj.prev** 場所を訊ねられたとき、前に近い物に移動するキー。標準設定は 'O'。

**getpos.menu** 場所を訊ねられて、目標の中を循環して移動するキーのいずれかを使っているとき、代わりにメニューを表示するかを切り替える。標準設定は '!'。

**getpos.moveskip** 場所を訊ねられて、高速移動のためにシフトを押しながらの移動や、meta-数字 キーでの移動を使っているとき、8 マス毎ではなく同じ記号を飛ばして移動する。標準設定は '\*'。

**getpos.filter** 場所を訊ねられて、目標の中を循環して移動するキーのいずれかを使っているとき、フィルタモードを切り替える。フィルタなし、視界内のみ、同じエリアのみを順に切り替える。標準設定は '"'。

**getpos.pick** 場所を訊ねられたとき、場所を選択し、おそらくさらなる情報を訊ねるためのキー。標準設定は '.'。

**getpos.pick.once** 場所を訊ねられたとき、場所を選択し、さらなる情報を訊ねるのを飛ばすキー。標準設定は ','。

**getpos.pick.quick** 場所を訊ねられたとき、場所を選択し、さらなる情報を訊ねるのを飛ばし、場所の問い合わせのループを終了するキー。標準設定は ';'。

**getpos.pick.verbose** 場所を訊ねられたとき、場所を選択し、訊ねられることなくさらなる情報を表示するキー。標準設定は ':'。

**getpos.self** 場所を訊ねられたとき、自分の位置に移動するキー。標準設定は '@'。

**getpos.unexplored.next** 場所を訊ねられたとき、次に近い未探索の場所に移動するキー。標準設定は 'x'。

**getpos.unexplored.prev** 場所を訊ねられたとき、前に近い未探索の場所に移  
動するキー。標準設定は 'x'。

**getpos.valid** 場所を訊ねられたとき、正当な目標の場所を表示するキー。標準  
設定は '\$'。

**getpos.valid.next** 場所を訊ねられたとき、次に近い正当な位置に移動するキー。  
標準設定は 'z'。

**getpos.valid.prev** 場所を訊ねられたとき、前に近い正当な位置に移動するキー。  
標準設定は 'Z'。

**nopickup** 物を拾わずに移動する前置キー。標準設定は 'm'。

**redraw** 画面を再描画するキー。標準設定は '^R'。

**redraw.numpad** 画面を再描画するキー。 *number\_pad* がオンの場合のみ。標  
準設定は '^L'。

**repeat** 前回のコマンドを繰り返すキー。標準設定は '^A'。

**reqmenu** 一部のコマンドからメニューを要求する前置キー。標準設定は 'm'。

**run** ある方向に走るための前置キー。標準設定は 'G'。

**run.nopickup** 途中にある物を拾わずにある方向に走るための前置キー。標準  
設定は 'M'。

**run.numpad** ある方向に走るための前置キー。 *number\_pad* がオンの場合のみ。  
標準設定は '5'。

**rush** ある方向に突進するための前置キー。標準設定は 'g'。

## メッセージ型の設定

メッセージがユーザー定義パターンにマッチングしたとき、メッセージをメッセージエリアに表示する方  
法を変更できる。

一般的に、メッセージ型を記述する設定ファイルエントリは次のようなものである:

```
MSGTYPE=type "pattern"
```

*type*                    どのようにメッセージを表示するか:  
**show** — 通常通りメッセージを表示する。  
**hide** — メッセージを表示しない。  
**stop** — more プロンプトでユーザーを待つ。  
**norep** — 1 度目はメッセージを表示するが、他のメッセージが来るまで再びは表示し  
ない。

*pattern*                マッチングするパターン。パターンは正規表現である必要がある。

次のものは *NetHack* の内部パターンマッチング機能を使ったメッセージ型の例である。

```
MSGTYPE=stop "You feel hungry."  
MSGTYPE=hide "You displaced *."
```

このように指定すると、“You feel hungry”というメッセージが現れた時は常に more プロンプトが表示され、“You displaced (何か).”にマッチングするメッセージは一切表示されない。

MSGTYPE 行の定義順は重要である;最後にマッチングした規則が使われる。一般的な場合を先に書き、例外をその後に書くこと。

## メニュー色の設定

一部のプラットフォームでは、メニュー行がユーザー定義パターンにマッチングしたとき、メニュー行に使う色を定義できる。現在のところ `tty`, `curses`, `win32tty`, `win32gui` インターフェースがこれに対応している。

一般的にメニュー色割り当てを記述する設定ファイルエントリは次のようなものである:

```
MENUCOLOR="pattern"=color&attribute
```

<i>pattern</i>	マッチングするパターン;
<i>color</i>	パターンがマッチングした行に使う色;
<i>attribute</i>	パターンがマッチングした行に使う属性。属性はオプションで、省略するなら、アンド記号も省略しなければならない。属性が定義されない場合、属性は使われない。

パターンは正規表現である必要がある。

指定できる色は: *black*, *red*, *green*, *brown*, *blue*, *magenta*, *cyan*, *gray*, *orange*, *light-green*, *yellow*, *light-blue*, *light-magenta*, *light-cyan*, *white*。そして他の色と同じ必要のない標準の前面色である *no-color*。

指定できる属性は: *none*, *bold*, *dim*, *underline*, *blink*, *inverse*。Normal は *none* の別名である。なお、プラットフォームはこれを好きなように解釈するかも知れない。

次のものは *NetHack* の内部パターンマッチング機能を使ったメニュー色の例である。

```
MENUCOLOR="* blessed *"=green
MENUCOLOR="* cursed *"=red
MENUCOLOR="* cursed *(being worn)"=red&underline
```

このように指定すると、“blessed”が含まれているメニュー行は緑色に、“cursed”の行は赤色に、“cursed”に引き続いて同じ行に“(being worn)”がある場合は赤色かつ下線付きで表示する。設定ファイル中に複数の MENUCOLOR エントリを書くことができ、メニュー行にマッチングした中で最後の MENUCOLOR 行がその行に使われる。

“uncursed”にマッチングするような色指定を意図しているなら、呪われていないことが分かっている全ての持ち物について“uncursed”と表示するための *implicit\_uncursed* オプションをおそらく設定した方がよいことに注意すること。

## 音声の設定

メッセージウィンドウに表示されたメッセージがユーザー定義のパターンと一致したときに特定の音声ファイルを鳴らすことができるプラットフォームもある。現在のところ、Qt, `win32tty`, `win32gui` がユーザー音声に対応している。

ユーザー音声とメッセージの関連付けに関する設定ファイル項目は以下のとおりである。

**SOUND**DIR 音声ファイルのあるディレクトリ

**SOUND** 音声ファイルとユーザーが指定するメッセージパターンを関連付ける項目。各 **SOUND** 項目は以下の項目に分解される。

**MSG** — メッセージウィンドウ指定 (3.6 ではただ一つだけに対応している)

**pattern** — マッチするパターン

**sound file** — 鳴らす音声ファイル

**volume** — 音声ファイルを鳴らすときの音量

このパターンは POSIX 拡張正規表現で指定する。

## ステータスハイライトの設定

あなたの入手した *NetHack* はステータスハイライトに対応してコンパイルされているかもしれない。その場合、ステータスディスプレイのフィールドの色や見た目を閾値を設定することでカスタマイズできる。ステータス色を定義する形式は:

`OPTION=hilite_status:field-name/behavior/color&attributes`

例えば、設定ファイルに次の行を書くと、ヒットポイントが 30 ヒットポイントが赤色で表示される:

`OPTION=hilite_status:hitpoints/<=30%/red/normal`

(この例は実際には `<=30%` で `red&normal` に、`>30%` で `no-color&normal` にする。)

もう一つの例として、次の行を設定ファイルに書くと、賢さが低下すると赤で表示され、上昇すると緑で表示される。

`OPTION=hilite_status:wisdom/down/red/up/green`

指定できる色は: black, red, green, brown, blue, magenta, cyan, gray, orange, light-green, yellow, light-blue, light-magenta, light-cyan, white。および *no-color*、ディスプレイの標準前景色; 黒や白やその他の色と同じである必要はない。

指定できる属性は: none, bold, dim, underline, blink, inverse。"Normal" は "none" の別名である; これらは他の属性と組み合わせて使うべきではない。

色と属性の両方を指定するには、それらを結合するために '&' を使う。複数の属性を指定するには、それらを結合するために '+' を使う。

例: `magenta&inverse+dim`。

なお、ディスプレイはその能力によっては特定の属性を置き換えたり無視したりするかもしれない、一般的に、これを好きなように解釈するかも知れないことに注意すること。例えば、ディスプレイシステムによっては太字を点滅にしたりその逆にしたりする。一方、他の属性が既に設定されている時に他の属性を要求すると、属性を合成するのではなく、以前の属性を置き換える。*nethack* は属性要求を (少なくとも *tty* インターフェースでは) 同時にではなく順々に発行するため、この様な状況を制御する唯一の方法は一つの属性として指定することである。

次のステータス項目を調整できる:

title	dungeon-level	experience-level
strength	gold	experience
dexterity	hitpoints	HD
constitution	hitpoints-max	time
intelligence	power	hunger
wisdom	power-max	carrying-capacity
charisma	armor-class	condition
alignment		score

疑似フィールド ‘characteristics’ は Str, Dex, Con, Int, Wis, Cha の六つ全てを一度に設定するのに使える。‘HD’ は ‘ヒットダイス (hit dice)’ で、変化したときに表示される経験レベルの概算である。‘experience’, ‘time’, ‘score’ はその他のオプション設定に依存して条件によっては表示される。

振る舞いの代わりに、‘condition’ は次の状態フラグを取れる: *stone, slime, strngl, foodpois, termill, blind, deaf, stun, conf, hallu, lev, fly, ride*。stone から termill の別名として ‘major\_troubles’ を、blind through hallu の別名として ‘minor\_troubles’ を、lev, fly, ride の別名として ‘movement’ を、そして全ての別名として ‘all’ を使える。

指定できる振る舞いは “always”, “up”, “down”, “changed”, 割合または絶対値の閾値、マッチングするテキストである。

**always** はそのフィールドの標準属性を設定する。

**up, down** は、フィールドの値が上昇または下降したときのためのフィールド属性を設定する。この属性は **statushilites** ターン後に元に戻る。

**changed** は、フィールドの値が変更されたときのためのフィールド属性を設定する。この属性は **statushilites** ターン後に元に戻る。(あるフィールドに、フィールドの値の変更に一致する、“changed” 規則と “up” または “down” 規則の両方がある場合、“up” または “down” が優先される。)

**percentage** は、フィールドの値がその百分率に一致したときのためのフィールド属性を設定する。0 から 100 までの数値に引き続いて ‘%’ (パーセント記号) で指定される。百分率に ‘<=’ または ‘>=’ が前置されている場合、値がその百分率よりも小さいまたは大きい場合にも一致する。厳密に小さいまたは大きい場合のみ一致するためには、‘<’ または ‘>’ を前置する。(数値制限はこのために少し緩められている: >-1% と <101% は許されている。) 四つのフィールドのみが百分率規則に対応している。“hitpoints” と “power” の百分率は直感的である; これらはそれぞれの最大値を基にしている。百分率によるハイライトは “*experience level*” と “*experience points*” (*showexp* オプションが有効の場合のみ) も許されている。これらの場合、百分率は現在の経験レベルの開始から次のレベルの開始までの進捗を基にしている。従って、レベル 2 が 20 ポイントで開始し、3 が 40 ポイントで開始する場合、30 ポイントは 50% であり 35 ポイントは 75% である。経験値については 100% は達成できない; レベルは上昇し、計算は次のレベルに対してリセットされるからである; しかし、=100% という規則は許され、次のレベルにちょうど 1 経験ポイントだけ足りない特殊な場合に一致する。

**absolute** 値はフィールド値がこの数値に一致した場合に属性を設定する。数値は 0 以上でなけ

ればならず (例外は “armor-class” で負数も許される)、オプションで ‘=’ を前置できる。代わりに数値の前に ‘<=’ または ‘>=’ が前置された場合、値がより小さいまたはより大きい場合にも一致する。‘<’ または ‘>’ を前置すると、厳密に小さいまたは大きい場合にのみ一致する。

`text` はフィールドの値がそのテキストに一致した場合に属性を設定する。テキストは “alignment”, “carrying-capacity”, “hunger”, “dungeon-level”, “title” にのみ利用できる。title の場合、職業のランク名のみがテストされる; キャラクタ名は無視される。

ゲーム内のオプションメニューは、設定ファイルでの正しい文法を決定する助けになるだろう。

オプション `statshilites` を 0 にすることでこの機能全体を無効にできる。

ハイライトの例:

```
OPTION=hilite_status: gold/up/yellow/down/brown
OPTION=hilite_status: characteristics/up/green/down/red
OPTION=hilite_status: hitpoints/100%/gray&normal
OPTION=hilite_status: hitpoints/<100%/green&normal
OPTION=hilite_status: hitpoints/<66%/yellow&normal
OPTION=hilite_status: hitpoints/<50%/orange&normal
OPTION=hilite_status: hitpoints/<33%/red&bold
OPTION=hilite_status: hitpoints/<15%/red&inverse
OPTION=hilite_status: condition/major/orange&inverse
OPTION=hilite_status: condition/lev+fly/red&inverse
```

## NetHack のシンボルの変更

NetHack は `symbol` ファイルからシンボル集合全体を読み込むことができる。

`symbol` ファイルから特定のシンボル集合を選択するために使われるオプションは:

`symset`           読み込みたいシンボル集合の名前を設定する。 `symbols`。

`roguesymset`       rogue レベルで表示するために読み込みたいシンボル集合の名前を設定する。

`SYMBOLS` と `ROGUESYMBOLS` の設定ファイルオプションを使っていくつかのシンボルを上書きすることもできる。シンボルは `name:value` の組で指定する。NetHack は `value` 文字列を伝統的な C の方法でエスケープする。つまり、‘\’ は次の文字をリテラルにする接頭辞である。従って、‘\’ は ‘\\’ と表現する必要がある。特殊接頭辞 ‘\m’ はシンボル値のメタビットを切り替え、‘^’ 接頭辞は引き続き文字を制御文字として扱う。

表 1: NetHack Symbols

標準設定	シンボル名	説明
	S_air	(air)
-	S_altar	(altar)
"	S_amulet	(amulet)
A	S_angel	(angelic being)
a	S_ant	(ant or other insect)
^	S_anti_magic_trap	(anti-magic field)
[	S_armor	(suit or piece of armor)

表 1: NetHack Symbols

標準設定	シンボル名	説明
[	S_armour	(suit or piece of armor)
^	S_arrow_trap	(arrow trap)
0	S_ball	(iron ball)
#	S_bars	(iron bars)
B	S_bat	(bat or bird)
^	S_bear_trap	(bear trap)
-	S_blcorn	(bottom left corner)
b	S_blob	(blob)
+	S_book	(spellbook)
)	S_boomleft	(boomerang open left)
(	S_boomright	(boomerang open right)
'	S_boulder	(boulder)
-	S_brcorn	(bottom right corner)
C	S_centaur	(centaur)
-	S_chain	(iron chain)
#	S_cloud	(cloud)
c	S_cockatrice	(cockatrice)
\$	S_coin	(pile of coins)
#	S_corr	(corridor)
-	S_crwall	(wall)
-	S_darkroom	(dark room)
^	S_dart_trap	(dart trap)
&	S_demon	(major demon)
*	S_digbeam	(dig beam)
>	S_dnladder	(ladder down)
>	S_dnstair	(staircase down)
d	S_dog	(dog or other canine)
D	S_dragon	(dragon)
;	S_eel	(sea monster)
E	S_elemental	(elemental)
/	S_explode1	(explosion top left)
-	S_explode2	(explosion top center)
\	S_explode3	(explosion top right)
	S_explode4	(explosion middle left)
	S_explode5	(explosion middle center)
	S_explode6	(explosion middle right)
\	S_explode7	(explosion bottom left)
-	S_explode8	(explosion bottom center)
/	S_explode9	(explosion bottom right)
e	S_eye	(eye or sphere)
^	S_falling_rock_trap	(falling rock trap)
f	S_feline	(cat or other feline)
^	S_fire_trap	(fire trap)
!	S_flashbeam	(flash beam)
%	S_food	(piece of food)
{	S_fountain	(fountain)
F	S_fungus	(fungus or mold)
*	S_gem	(gem or rock)
*	S_ghost	(ghost)
H	S_giant	(giant humanoid)
G	S_gnome	(gnome)
,	S_golem	(golem)
	S_grave	(grave)
g	S_gremlin	(gremlin)

表 1: NetHack Symbols

標準設定	シンボル名	説明
-	S_hbeam	(wall)
#	S_hcdbridge	(horizontal raised drawbridge)
+	S_hcdoor	(closed door)
.	S_hodbridge	(horizontal lowered drawbridge)
	S_hodoor	(open door)
^	S_hole	(hole)
@	S_human	(human or elf)
h	S_humanoid	(humanoid)
-	S_hwall	(horizontal wall)
.	S_ice	(ice)
i	S_imp	(imp or minor demon)
I	S_invisible	(invisible monster)
J	S_jabberwock	(jabberwock)
j	S_jelly	(jelly)
k	S_kobold	(kobold)
K	S_kop	(Keystone Kop)
^	S_land_mine	(land mine)
}	S_lava	(molten lava)
l	S_leprechaun	(leprechaun)
^	S_level_teleporter	(level teleporter)
L	S_lich	(lich)
y	S_light	(light)
#	S_litcorr	(lit corridor)
:	S_lizard	(lizard)
\	S_lslant	(wall)
^	S_magic_portal	(magic portal)
^	S_magic_trap	(magic trap)
m	S_mimic	(mimic)
]	S_mimic_def	(mimic)
M	S_mummy	(mummy)
N	S_naga	(naga)
.	S_ndoor	(doorway)
n	S_nymph	(nymph)
O	S_ogre	(ogre)
o	S_orc	(orc)
p	S_piercer	(piercer)
^	S_pit	(pit)
#	S_poisoncloud	(poison cloud)
^	S_polymorph_trap	(polymorph trap)
}	S_pool	(water)
!	S_potion	(potion)
P	S_pudding	(pudding or ooze)
q	S_quadraped	(quadraped)
Q	S_quantmech	(quantum mechanic)
=	S_ring	(ring)
‘	S_rock	(boulder or statue)
r	S_rodent	(rodent)
^	S_rolling_boulder_trap	(rolling boulder trap)
.	S_room	(floor of a room)
/	S_rslant	(wall)
^	S_rust_trap	(rust trap)
R	S_rustmonst	(rust monster or disenchanter)
?	S_scroll	(scroll)
#	S_sink	(sink)

表 1: NetHack Symbols

標準設定	シンボル名	説明
^	S_sleeping_gas_trap	(sleeping gas trap)
S	S_snake	(snake)
s	S_spider	(arachnid or centipede)
^	S_spiked_pit	(spiked pit)
^	S_squeaky_board	(squeaky board)
0	S_ss1	(magic shield 1 of 4)
#	S_ss2	(magic shield 2 of 4)
@	S_ss3	(magic shield 3 of 4)
*	S_ss4	(magic shield 4 of 4)
^	S_statue_trap	(statue trap)
	S_stone	(solid rock or unexplored terrain or dark part of a room)
]	S_strange_obj	(strange object)
-	S_sw_bc	(swallow bottom center)
\	S_sw_bl	(swallow bottom left)
/	S_sw_br	(swallow bottom right )
	S_sw_ml	(swallow middle left)
	S_sw_mr	(swallow middle right)
-	S_sw_tc	(swallow top center)
/	S_sw_tl	(swallow top left)
\	S_sw_tr	(swallow top right)
-	S_tdwall	(wall)
^	S_teleportation_trap	(teleportation trap)
\	S_throne	(opulent throne)
-	S_tlcorn	(top left corner)
	S_tlwall	(wall)
(	S_tool	(useful item (pick-axe, key, lamp...))
^	S_trap_door	(trap door)
t	S_trapper	(trapper or lurker above)
-	S_trcorn	(top right corner)
#	S_tree	(tree)
T	S_troll	(troll)
	S_trwall	(wall)
-	S_tuwall	(wall)
U	S_umber	(umber hulk)
u	S_unicorn	(unicorn or horse)
<	S_upladder	(ladder up)
<	S_upstair	(staircase up)
V	S_vampire	(vampire)
	S_vbeam	(wall)
#	S_vcdbridge	(vertical raised drawbridge)
+	S_vcdoor	(closed door)
.	S_venom	(splash of venom)
^	S_vibrating_square	(vibrating square)
.	S_vodbridge	(vertical lowered drawbridge)
-	S_vodoor	(open door)
v	S_vortex	(vortex)
	S_vwall	(vertical wall)
/	S_wand	(wand)
}	S_water	(water)
)	S_weapon	(weapon)
"	S_web	(web)
w	S_worm	(worm)
~	S_worm_tail	(long worm tail)

表 1: NetHack Symbols

標準設定	シンボル名	説明
W	S_wraith	(wraith)
x	S_xan	(xan or other extraordinary insect)
X	S_xorn	(xorn)
Y	S_yeti	(apelike creature)
Z	S_zombie	(zombie)
z	S_zruty	(zruty)
	S_pet_override	(any pet if ACCESSIBILITY=1 is set)
	S_hero_override	(hero if ACCESSIBILITY=1 is set)

注意:

この表のいくつかのシンボルは空白に見える。これらはスペース文字である; 例外は、標準設定値がなく、“sysconf” ファイルで有効にされた場合にのみ使われる S\_pet\_override と S\_hero\_override である。

S\_rock は語弊のある名前である; 岩と石は S\_gem である。石像と巨岩は参照される岩であるが、バージョン 3.6.0 から、石像はモチーフとなっている怪物として表示される。従って、S\_rock は巨岩のためだけに使われ、これがより限定的な S\_boulder で上書きされると、全く使われないことになる。

## 視覚障害者が NetHack をプレイするための設定

NetHack は洞窟の地図を作るのに標準の ASCII キャラクターだけを使うように設定することもできる。これにより、MS-DOS 版の NetHack は音声合成または点字技術を用いれば目の不自由な人でも完全にプレイ可能である。プレイヤーはこれらの画面読み取りシステムに関する十分な知識が必要で、上下左右に一文字ずつ移動する方法を知らなければならないだろう。画面読み取りシステムの検索能力を知ることかなり価値がある。プレイする前にこのガイドブックを十分に読めば、画面のレイアウトがどんな感じかわかるだろう。また、PC カーソルの位置を突き止めることができる必要がある。これは自分のキャラクターの位置を示している。‘@’ は他のヒューマノイドを表わすのにも用いられるので、自分の位置を探すのに ‘@’ を検索するだけでは不十分である。さらに、あなたが見ている位置と PC カーソルの位置の物理的な行と桁を知らせる機能が必要だろう。この機能はスクリーン全体にどのようなアイテムがあるかを判別するのに便利である。NetHack は、ゲームメッセージをテキスト/スピーチシンセサイザーなどの外部プログラムに送信するサポート付きでコンパイルすることもできる。拡張コマンド “#version” に “external program as a message handler” と表示されている場合、NetHack はこの機能付きでコンパイルされている。Linux やその他の POSIX システムのソースから NetHack をコンパイルする際には、これを有効にするには MSGHANDLER を定義する。この機能を使用するには、環境変数 NETHACK\_MSGHANDLER を実行可能ファイルに設定する; 実行可能ファイルは、ゲームメッセージをプログラムの唯一の引数として実行される。慣れたユーザーなら defaults.nh を修正することは難しくないだろうが、初心者は少しひるんでしまう仕事かもしれない。NetHack の全ての公式配布のシンボルファイルには NHAccess と呼ばれる symset が含まれている。設定ファイルでこの symset を選択することにより、目の不自由な人がプレイできるような設定になる。ファイルを修正することやゲームそのものに慣れたなら、好みに合うように設定ファイルの SYMBOLS= と ROGUESYMBOLS= で設定を変更するとよいだろう。全てのペットに一貫したシンボルを強制する特殊シンボル S\_pet\_override と、sysconf ファイルで accessibility が有効の時にプレイヤーキャラクターに特別なシンボルを強制する特殊シンボル S\_hero\_override については前述の節を参照のこと。

ゲームのアクセシビリティの改善に最も影響を与える設定は以下のものである:

<i>symset:NHAccess</i>	視力障害者プレイヤーに適したシンボル集合を読み込む。
<i>roguesymset:NHAccess</i>	視力障害者プレイヤーに適した rogue レベルのためのシンボル集合を読み込む。
<i>menustyle:traditional</i>	これは音声合成システムを利用する助けになる。
<i>nomenu_overlay</i>	クリアされた画面に左寄せでメニューを表示する。
<i>number_pad</i>	多くの発声プログラムは number-pad を画面の確認に用いる。この場合、number_pad オプションをオフにして、伝統的な Rogue 風コマンドを使うこと。
<i>autodescribe</i>	目標を決めるときにカーソルの下の地形を自動的に表現する。
<i>mention_walls</i>	壁に向かって歩いたり長距離移動コマンドが中断されたりしたときにメッセージを表示する。
<i>whatis_coord:compass</i>	カーソルで目標を決めるとき、自分の位置からのカーソルの相対座標を表現する。
<i>whatis_filter:area</i>	カーソルで目標を決めるとき、同じエリア (例えば同じ部屋、同じ通路) のみになるように可能な場所をフィルタリングする。
<i>whatis_moveskip</i>	カーソルで目標を決めて高速移動を使うとき、一度に 8 マスずつ動くのではなく、同じ記号を飛ばす。
<i>nostatus_updates</i>	スクリーンリーダーが画面下部のステータス行を読むとき、その更新を抑制する。同じ情報は #attributes コマンドでも得られる。

## システム管理者のためのグローバルな設定

*NetHack* が SYSCF オプション付きでコンパイルされている場合、システム管理者はグローバル設定を設定すべきである; これは (前述した) 伝統的なユーザー毎の設定ファイルと同じ形式のファイルである。

このファイルは *sysconf* という名前で、その他の *NetHack* 補助ファイルと同じディレクトリに置かれる。このファイルで認識されるオプションは以下の通りである。設定されてなかったオプションは (あなたのシステムでは適切ではないかもしれない) コンパイル時のデフォルトが使われる。

<i>WIZARDS</i>	(一般的にウィザードモードと呼ばれる) デバッグモードでプレイすることを許されたユーザー名の空白区切りのリスト。値が単一のアスタリスク (*) の場合は誰でもデバッグモードでゲームを開始できる。
<i>SHELLERS</i>	シェルエスケープコマンド ('!') の使用を許されたユーザーのリスト。文法は <i>WIZARDS</i> と同じ。
<i>EXPLORERS</i>	探検モードを使うことを許されたユーザーのリスト。文法は <i>WIZARDS</i> と同じ。
<i>MAXPLAYERS</i>	同時にプレイ出来るゲームの数。
<i>SUPPORT</i>	ローカルサポートを受ける方法を示す文字列 (標準設定値はなし)。
<i>RECOVER</i>	このシステムでゲームを復旧する方法を示す文字列 (標準設定値はなし)。
<i>SEDUCE</i>	0 と 1 でそれぞれ <i>SEDUCE</i> オプションを無効/有効にする。無効にすると、インキュバスとサキュバスはニンフのように振る舞う。
<i>CHECK_PLNAME</i>	これを 1 にすると、 <i>EXPLORERS</i> , <i>WIZARDS</i> , <i>SHELLERS</i> はユーザーのログイン名ではなくプレイヤー名をチェックする。

*CHECK\_SAVE\_UID* 0 と 1 でそれぞれ、(復元するユーザーがセーブしたユーザーと同じかを検証するために) セーブファイルの UID (識別番号として使われる) チェックを無効/有効にする。

以下のオプションは score ファイルに影響する:

*PERSMAX* 一人のエントリの最大数。  
*ENTRYMAX* score ファイルのエントリの最大数。  
*POINTSMIN* score ファイルにエントリを記録する最低得点。  
*PERS\_IS\_UID* score ファイル上でプレイヤーを認識するのに、0 ではユーザー名、1 では数値のユーザー ID を使う。  
*ACCESSIBILITY* プレイヤーが自分自身の設定ファイルで *S\_pet\_override* and *S\_hero\_override* シンボルを設定することを、0 では無効に、1 では有効にする。  
*PORTABLE\_DEVICE\_PATHS* Windows OS のみ; 全ての外部ファイルを検索し、全ての出力ファイルを書き込むディレクトリを、標準の位置ではなく一ヶ所にする。  
*DUMPLOGFILE* ゲーム終了時のダンプログを保存するファイル名。これを定義していないとダンプログは作成されない。ゲームが DUMPLOG 付きでコンパイルされている場合にのみ利用可能。次のプレースフォルダが利用できる: % — リテラルな ‘%’  
%v — バージョン (例: “3.6.3-0”)  
%u — ゲーム UID  
%t — ゲーム開始時刻、UNIX タイムスタンプ形式  
%T — 現在時刻、UNIX タイムスタンプ形式  
%d — ゲーム開始時刻、YYYYMMDDhhmmss 形式  
%D — 現在時刻、YYYYMMDDhhmmss 形式  
%n — プレイヤー名  
%N — プレイヤー名の最初の文字

## 10 得点

*NetHack* は設定によって、あなたの機械における高得点の一覧または高得点者の一覧を作成する。後者の設定になっている場合、ゲームを最後まで終えた場合を除きそのマシンの 1 つのアカウントにつき 1 つだけしかこの一覧には載らない。あなたがこの一覧に載っている他の誰かより高い得点を取ったとき、もしくはあなたの前の得点より高い得点を取ったときにのみ、その一覧のしかるべき位置にあなたの名前が載る。得点が何個まで載るかもコンパイル時に設定できる。

得点は主に経験値、戦利品、到達した階とゲームの終わり方に基づいている。ゲームを放棄した場合には持っている金を全額手にして脱出できる。しかし恐怖の迷宮の中で殺された場合には、亡骸が発見されると持っている金の 90% だけがギルドに伝えられる (冒険者は亡骸を発見した場合には手数料を取ることが知られている)。従って怪物にさらに一撃を加えて運あれば生き残らんとするか、放棄してその時点で持っている物で終了するかはあなたが決めることである。放棄した場合はすべての金を手にすることができる。しかし戦って生き残ればさらに多くを得ることができるかも知れない。

単に現時点での高得点者・高得点の一覧を見たい場合にはほとんどのバージョンでは

```
nethack -s all
```

と入力すればよい。

## 11 探検モード

*NetHack* は複雑で難しいゲームである。初心者は生き残るすべを知らないことに気づき、恐怖におじけづいてしまうかもしれない。しかし恐れることはない。冒険する洞窟に「探検」もしくは「発見」モードがある場合には、高得点の一覧に載らないというところに足らないデメリットと引き替えに前のセーブファイルを残しておけたり不死身となれたりするのである。

探検モードに入るには 2 通りの方法がある。1 つはゲームの開始時に `-x` スイッチを付けるか `playmode:explore` オプションを指定することである。もう 1 つはゲームのプレイ中に `#exploremode` 拡張コマンドを入力することである。探検モードで新しいゲームを開始すると、初期装備として願いの杖が追加される; プレイ中に切り替えた場合は追加されない。探検モードで得られる他の利点は大胆な読者諸君自ら確かめて欲しい。

## デバッグモード

ウィザードモードとも呼ばれるデバッグモードは、この概説と、コマンド説明で挙げられている様々な「デバッグモード専用」コマンドの他には文書化されていない。これはキャラクターに神のような力を与えるためではなく、プログラムの問題を調べることを意図したもので、デバッグを試みるプレイヤーはこれをどのように使うかが分かることを想定している。これは `-D` コマンドラインスイッチか `playmode:debug` オプションでゲームを始めることで開始される。

システムによっては、プレイヤーは、デバッグモードを使うことを許されている特定のユーザー名でログインしなければならない; それ以外では、キャラクターは特定のキャラクター名を使わなければならない(しかし職業はどれでもよい; “ウィザード” という名前と職業の *Wizard* は何の関係もない)。デバッグモードを許されていなかったり有効でない場合にデバッグモードを使おうとすると、代わりに探検モードにフォールバックする。

## 12 クレジット

最初の *hack* というゲームは、バークレイ版 UNIX のゲーム *rogue* をモデルとしている。このドキュメントの大部分は Michael C. Toy と Kenneth C. R. C. Arnold の手になる *A Guide to the Dungeons of Doom*(運命の洞窟への招待) を図々しくも盗用したものである。また一部は Ken Arromdee による *Further Exploration of the Dungeons of Doom*(運命の洞窟の深部への探検) から採っている。

*NetHack* は文字どおり何十人もの人々の手によって完成された。以下はゲームの開発の過程での主な出来事である。

*Jay Fenlason* は *Kenny Woodland*, *Mike Thome*, *Jon Payne* らの助けを得て、最初の *Hack* を書いた。

*Andries Brouwer* はプログラムを大幅に書き換えて *Hack* を元とはかなり異なったゲームに仕立て上げ、UNIX マシン用の (少なくとも)3 バージョン (1.0.1, 1.0.2, 1.0.3) を Usenet で発表した。

*Don G. Kneller* は *Hack 1.0.3* を Microsoft C を使用して MS-DOS へ移植し *PC Hack 1.01e* を作成した。バージョン 1.03g では DEC Rainbow のグラフィックをサポートし、続いて少なくともあと 4 つのバージョン (3.0、3.2、3.51、3.6) を作成した。

*R. Black* は *PC Hack 3.51* を Lattice C を使用して Atari 520/1040ST へ移植し *ST Hack 1.03* を作成した。

*Mike Stephenson* はこれらのいろいろなバージョンを再び 1 つに統合し、多くの追加要素と合わせて *NetHack* バージョン 1.4 を作成した。さらに彼は何千人もの人々に作業を分担して *NetHack 1.4* の拡張とデバッグを行い、*NetHack* バージョン 2.2 と 2.3 を発表した。

さらにその後、Mike は *Ken Arromdee*, *Jean-Christophe Collet*, *Steve Creps*, *Eric Hendrickson*, *Izchak Miller*, *Eric S. Raymond*, *John Rupley*, *Mike Threepoint*, *Janet Walz* を含むチームを率いてゲームの大幅な書き直しを行ない、*NetHack 3.0c* を作成した。

*NetHack 3.0* は *Eric R. Smith* によって Atari へ、*Timo Hakulinen* によって OS/2 へ、*David Gentzel* によって VMS へ移植された。この 3 人と *Kevin Darcy* はその後主力開発チームに加わり、3.0 のその後のいくつかの改訂版を作成した。

*Olaf Seibert* は *NetHack 2.3* と 3.0 を Amiga へ移植した。*Norm Meluch*, *Stephen Spackman*, *Pierre Martineau* は *PC NetHack 3.0* のためのオーバーレイルーチンを設計した。*Johnny Lee* は *NetHack 3.0* を Macintosh へ移植した。彼らは他のさまざまな洞窟の主たちとともに PC、Macintosh、Amiga の移植版の拡張を続け、3.0 のその後のいくつかの改訂版を作成した。

*Mike Stephenson* をリーダーとし、*Izchak Miller* と *Janet Walz* の助けによって *Ken Arromdee*, *David Cohrs*, *Jean-Christophe Collet*, *Kevin Darcy*, *Matt Day*, *Timo Hakulinen*, *Steve Linhart*, *Dean Luick*, *Pat Rankin*, *Eric Raymond*, *Eric Smith* を含む制作チームが 3.0 の徹底的な見直しを行なった。彼らはゲームデザインを再構築し、コードの大部分を書き直した。複合ダンジョン、新しいディスプレイ、それぞれのキャラクター毎の特別のクエスト、新しいエンドゲームとその他の多くの新しい要素を追加し、*NetHack 3.1* を制作した。

*Ken Lorber*, *Gregg Wonderly*, *Greg Olson* は *Richard Addison*, *Mike Passaretti*, *Olaf Seibert* の助けを借りて Amiga 用 *NetHack 3.1* を作成した。

*Norm Meluch* と *Kevin Smolkowski* は *Carl Schelin*, *Stephen Spackman*, *Steve VanDevender*, *Paul Winner* の助けを借りて *NetHack 3.1* を PC に移植した。

*Jon W{tte* と *Hao-yang Wang* は *Ross Brown*, *Mike Engber*, *David Hairston*, *Michael Hamel*, *Jonathan Handler*, *Johnny Lee*, *Tim Lennan*, *Rob Menke*, *Andy Swanson* の助けを借りて MPW 用の Macintosh 用 *NetHack 3.1* を作成した。その作成中に、*Bart House* は Think C 用を加えた。

*Timo Hakulinen* は *NetHack 3.1* を OS/2 に移植した。*Eric Smith* は *NetHack 3.1* を Atari に移植した。*Pat Rankin* は *Joshua Delahunty* の助けを借りて VMS 版 *NetHack 3.1* を作成した。*Michael Allison* は *NetHack 3.1* を Windows NT に移植した。

*Dean Luick* は *David Cohrs* の助けを借りて X11 用 *NetHack3.1* を作成した。*Warwick Allison* は Atari 用に *NetHack* のタイル版を作成した。彼は後にタイルを開発チームに寄贈し、タイルサポートは他のプラットフォームにも追加された。

*Michael Allison, Ken Arromdee, David Cohrs, Jessie Collet, Steve Creps, Kevin Darcy, Timo Hakulinen, Steve Linhart, Dean Luick, Pat Rankin, Eric Smith, Mike Stephenson, Janet Walz, Paul Winner* からなる 3.2 の開発チームは バージョン 3.2 を 1996 年 4 月にリリースした。

バージョン 3.2 は開発チームが結成されてから 10 周年のものであった。彼らのゲームへの貢献について記すと、13 人の初期の開発チーム全員はこのリリースの開始時の最初の時点までチームに残っていた。が、3.1.3 から 3.2 の間に、開発チームの創始者の一人である *Izchak Miller* 博士が癌と診断され、亡くなった。このリリースは開発チームおよび移植チームから彼に捧げられた。

*NetHack* 3.1 と 3.2 の時代に、ゲームに熱狂した何人かがゲームに自分自身の変更を加えた「亜種」を公に発表した。

*Tom Proudfoot* と *Yuval Oren* は *NetHack++* を作成し、これはすぐに *NetHack--* に改名された。これとは独立に、*Stephen White* は *NetHack Plus* を作成した。*Tom Proudfoot* は後に *NetHack Plus* と *NetHack--* を統合し、*SLASH* を作成した。*Larry Stewart-Zerba* と *Warwick Allison* は呪文詠唱システムを改良した *Wizard Patch* を作成した。*Warwick Allison* は *NetHack* を Qt インターフェースを使うように変更した。

*Warren Cheung* は *SLASH* と *Wizard Patch* を統合して *Slash'EM* を作成し、*Kevin Hugo* の力を借りて、多くの要素を追加した。*Kevin* は後に開発チームに加わり、これらのアイデアを *NetHack* 3.3 に統合した。

3.2 の最終バージョンはバグ修正版である 3.2.3 である。これは 2000 年問題対策のために 3.3.0 と同時に 1999 年 12 月に発表された。

*Michael Allison, Ken Arromdee, David Cohrs, Jessie Collet, Steve Creps, Kevin Darcy, Timo Hakulinen, Kevin Hugo, Steve Linhart, Ken Lorber, Dean Luick, Pat Rankin, Eric Smith, Mike Stephenson, Janet Walz, Paul Winner* からなる 3.3 の開発チームは 1999 年 12 月に 3.3.0 を、2000 年 8 月に 3.3.1 をリリースした。

バージョン 3.3 は多くの新要素を提供した。まず、種族と職業を分離した最初のバージョンである。職業としてのエルフは削除され、種族の一つとなった。また、ドワーフ、ノーム、オークが種族として追加された。モンクとレンジャーの二つが職業として追加された。さらに、動物に乗ることが出来る最初のバージョンでもあり、発見された全てのバグをウェブサイトで公表した最初のバージョンでもある。バグリストが順調に伸びていったにも関わらず、3.3 は 1 年半続いたことで十分安定していたことを証明した。

3.4 開発チームは *Michael Allison, Ken Arromdee, David Cohrs, Jessie Collet, Kevin Hugo, Ken Lorber, Dean Luick, Pat Rankin, Mike Stephenson, Janet Walz, and Paul Winner* で開始され、*NetHack* 3.4.0 リリース直前の 2002 年 3 月に *Warwick Allison* が加わった。

バージョン 3.3 と同様に、様々な人々が *NetHack* の色々なプラットフォームへの移植を行なってくれた。*Pat Rankin* は 3.4 の VMS への移植を担当した。

*Michael Allison* は MS-DOS 版の *NetHack* 3.4 を管理している。*Paul Winner* と *Yitzhak Sapir* が手伝っている。

*Dean Luick, Mark Modrall, Kevin Hugo* は 3.4 の Macintosh への移植と拡張を担当した。

*Michael Allison, David Cohrs, Alex Kompel, Dion Nicolaas, Yitzhak Sapir* は 3.4 の Microsoft Windows プラットフォームへの移植と拡張を担当した。*Alex Kompel* は Windows 版への新しいグラフィックインターフェースを提供した。*Alex Kompel* はまた、3.4.1 の Windows CE 版を提供した。

*Ron Van Iwaarden* は過去いくつかのリリースの OS/2 版 *NetHack* の唯一の管理者だった。残念ながら Ron の最後の OS/2 マシンは 2006 年初頭に動かなくなった。近年まで OS/2 で *NetHack* を生き続けさせたことにことについて Ron に非常に感謝する。

Amiga への移植は *Janne Salmijärvi* が 3.3.1 で復活させた後、*Janne Salmijärvi* と *Teemu Suikki* が 3.4 への保守と拡張を行った。

3.3.1 で復活した Atari 版は *Christian “Marvin” Bressler* が 3.4 を保守した。

2003 年 12 月の *NetHack* 3.4.3 のリリースは、長いリリース中断の始まりとなった。3.4.3 は、コミュニティによって 10 年以上楽しまれ続けられたことで非常に安定したバージョンであることが示された。3.4.3 の間、*The NetHack Development Team* は水面下でゆっくりと静かにゲームに対して作業を続けていた。同じ頃、*NetHack* コミュニティにいくつかの新しいバリエーションが出現した。特に、Derek S. Ray による *sporkhack*、Patric Mueller による *unnethack*、元々は Daniel Thaler で、それからは Alex Smith による *nitrohack* およびその後継、Tung Nguyen による *Dynahack* である。これらのバリエーションの一部は開発と保守が続けられ、今でもコミュニティで楽しまれている。

2014 年 9 月に、開発中のコードの中間的なスナップショットが他のグループによって公にされた。コードは作業中のもので、適切なリリースのためのデバッグプロセスを経ていないものであったので、コードスナップショットに使われていたバージョン番号は欠番として、公式な *NetHack* リリースでは使わないことに決定した。*NetHack Development Team* の公式 *nethack.org* web サイトにアナウンスが投稿され、公式リリースバージョンとして 3.4.4, 3.5, 3.5.0 は使わないことを示した。

2015 年 1 月、*NetHack* 3.6 のリリースのための準備が開始された。

最終的に 3.6.0 としてリリースされるものの開発の初期では、開発チームは *Warwick Allison, Michael Allison, Ken Arromdee, David Cohrs, Jessie Collet, Ken Lorber, Dean Luick, Pat Rankin, Mike Stephenson, Janet Walz, Paul Winner* で構成されていた。2015 年初頭、3.6.0 リリースに先だって、新しいメンバーである *Sean Hunt, Pasi Kallinen, Derek S. Ray* が *NetHack* 開発チームに加わった。

3.6.0 の開発の終了近くに、one of the significant inspirations for many of the humorous and fun features found in the game, author Terry Pratchett, passed away. *NetHack* 3.6.0 は彼へ捧げられた。

3.6.0 は 2015 年 12 月にリリースされ、3.4.3 リリースから開発チームによってなされた作業と、いくつかの愛好者コミュニティのパッチがマージされた。多くのバグが修正され、一部のコードが再構成された。

*NetHack Development Team* 及び *Steve VanDevender* と *Kevin Smolkowski* は *NetHack* 3.6 が様々な Unix の亜種で動作し続けるようにし、また X11 インターフェースを管理している。

*Ken Lorber, Haoyang Wang, Pat Rankin, Dean Luick* は *NetHack* 3.6 の Mac OSX 版を管理している。

*Michael Allison, David Cohrs, Bart House, Pasi Kallinen, Alex Kompel, Dion Nicolaas, Derek S. Ray* and *Yitzhak Sapir* は *NetHack* 3.6 の Microsoft Windows 版を管理している。

*Pat Rankin* は、アクセスが制限されながら VMS 版を *NetHack* 3.6 で動作させ続けようとした。Kevin Smolkowski はこれを、Alpha と Integrity (またの名を Itanium またの名を IA64) での最新版の OpenVMS (記述時点で V8.4) に更新、テストしたが、VAX 版ではない。

*Ray Chason* は 3.6 の msdos 版を復活させ、必要な更新をコミュニティに寄贈した。

2018 年 4 月下旬、3.6.0 からの数百のバグ修正と、いくつかの新しい機能がまとめられ、NetHack 3.6.1 としてリリースされた。3.6.1 リリース時点での *NetHack Development Team* は、*Warwick Allison, Michael Allison, Ken Arromdee, David Cohrs, Jessie Collet, Pasi Kallinen, Ken Lorber, Dean Luick, Patric Mueller, Pat Rankin, Derek S. Ray, Alex Smith, Mike Stephenson, Janet Walz, Paul Winner* からなる。

2019 年 5 月初旬、更なる 320 のバグ修正といくつかの拡張と、寄贈された curses ウィンドウ対応を含んで、3.6.2 としてリリースされた。

数十年にわたって移植チームとしてゲームに貢献してきた *Bart House* は、2019 年 5 月に *NetHack Development Team* に加わった。

NetHack 3.6.3 は、NetHack 3.6.2 から 190 以上のバグ修正を含んで、2019 年 12 月 5 日にリリースされた。

NetHack 3.6.4 は、セキュリティ修正といくつかのバグ修正を含んで、2019 年 12 月 18 日にリリースされた。

NetHack 3.6.5 は、いくつかのセキュリティ修正とバグ修正を含んで、2020 年 1 月 27 日にリリースされた。

NetHack 3.6.6 は、セキュリティ修正といくつかのバグ修正を含んで、2020 年 3 月 8 日にリリースされた。

*Ken Lorber* によって管理されている *NetHack* 公式 web サイトは: <https://www.nethack.org/>.

## 謝辞

*NetHack* コミュニティに成り代わって、[nethack.alt.org](https://nethack.alt.org) で公開 NetHack サーバーを提供してくれていることについて再度 *M. Drew Streib* と *Pasi Kallinen* に感謝する。[hardfought.org](https://hardfought.org) に関して *Keith Simpson* と *Andy Thomson* に感謝する。*Junethack, The November NetHack Tournament* や以前の (今はないが忘れられない) *devnull.net* のような、年次 *NetHack* トーナメントに時間と労力を使ってくれた無名の洞窟の主全てに感謝する。

## 洞窟の主

時々ネットワークの世界のどうしようもない連中が、ゲームの改良の手助けをしようとしてとりわけ興味をそそるような修正を送ってよこす。*NetHack Development Team* は、ときにはこういった悪党のうちでも最も邪悪な連中の名前を洞窟の主たちの一覧としてここに記すのだ。

Adam Aronow	J. Ali Harlow	Mikko Juola
Alex Kompel	Janet Walz	Nathan Eady
Alex Smith	Janne Salmijärvi	Norm Meluch
Andreas Dorn	Jean-Christophe Collet	Olaf Seibert
Andy Church	Jeff Bailey	Pasi Kallinen
Andy Swanson	Jochen Erwied	Pat Rankin
Andy Thomson	John Kallen	Patric Mueller
Ari Huttunen	John Rupley	Paul Winner
Bart House	John S. Bien	Pierre Martineau
Benson I. Margulies	Johnny Lee	Ralf Brown
Bill Dyer	Jon W{tte	Ray Chason
Boudewijn Waijers	Jonathan Handler	Richard Addison
Bruce Cox	Joshua Delahunty	Richard Beigel
Bruce Holloway	Karl Garrison	Richard P. Hughey
Bruce Mewborne	Keizo Yamamoto	Rob Menke
Carl Schelin	Keith Simpson	Robin Bandy
Chris Russo	Ken Arnold	Robin Johnson
David Cohrs	Ken Arromdee	Roderick Schertler
David Damerell	Ken Lorber	Roland McGrath
David Gentzel	Ken Washikita	Ron Van Iwaarden
David Hairston	Kevin Darcy	Ronnen Miller
Dean Luick	Kevin Hugo	Ross Brown
Del Lamb	Kevin Sitze	Sascha Wostmann
Derek S. Ray	Kevin Smolkowski	Scott Bigham
Deron Meranda	Kevin Sweet	Scott R. Turner
Dion Nicolaas	Lars Huttar	Sean Hunt
Dylan O'Donnell	Leon Arnott	Stephen Spackman
Eric Backus	M. Drew Streib	Stefan Thielscher
Eric Hendrickson	Malcolm Ryan	Stephen White
Eric R. Smith	Mark Gooderum	Steve Creps
Eric S. Raymond	Mark Modrall	Steve Linhart
Erik Andersen	Marvin Bressler	Steve VanDevender
Fredrik Ljungdahl	Matthew Day	Teemu Suikki
Frederick Roeber	Merlyn LeRoy	Tim Lennan
Gil Neiger	Michael Allison	Timo Hakulinen
Greg Laskin	Michael Feir	Tom Almy
Greg Olson	Michael Hamel	Tom West
Gregg Wonderly	Michael Sokolov	Warren Cheung
Hao-yang Wang	Mike Engber	Warwick Allison
Helge Hafting	Mike Gallop	Yitzhak Sapir
Irina Rempt-Drijfhout	Mike Passaretti	
Izchak Miller	Mike Stephenson	