

次の問2から問7までの6問については、この中から4問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、5問以上マークした場合には、はじめの4問について採点します。

問2 言語処理系に関する次の記述を読んで、設問1～3に答えよ。

言語処理系とは、プログラム言語（以下、言語という）の文法に従って記述された原始プログラムを読み取り、目的とする計算機で実行するための変換をするか、又は逐次解釈しながら実行するためのソフトウェアである。

設問1 言語処理系の種類には、インタプリタやコンパイラなどがある。インタプリタによる実行では、原始プログラムを入力して、逐次解釈しながら実行する（以下、インタプリタ方式という）。コンパイラによる実行では、原始プログラムを入力して、目的とする計算機で実行できるプログラム（以下、実行形式プログラムという）に変換し、その実行形式プログラムを実行する（以下、コンパイラ方式という）。インタプリタ方式とコンパイラ方式を比較したとき、コンパイラ方式の利点として最も適切な答えを、解答群の中から選べ。

解答群

- ア 原始プログラムを対話的に確認しながら実行できる。
- イ 原始プログラムを変更して、直ちに実行できる。
- ウ 実行途中に異常が発生した場合、原始プログラムのどこでどのような原因によって異常が発生したのかを確認しやすい。
- エ 目的とする計算機に対応して、実行時間を短縮するための最適化が図れる。

設問2 次の記述中の に入れる最も適切な答えを、解答群の中から選べ。

原始プログラムを、ソフトウェアによって仮想的に構築した計算機（以下、仮想計算機という）で実行できるコード（以下、中間コードという）に変換し、仮想計算機上で中間コードをインタプリタ方式で実行する方法がある。言語 X で記述された原始プログラムを中間コードに変換し、仮想計算機 V のインタプリタで実行する例を、図 1 に示す。ここで、異なる OS やハードウェア上で動作する仮想計算機 V を用意することによって、 a 中間コードの形でプログラムを配布することが可能となる。

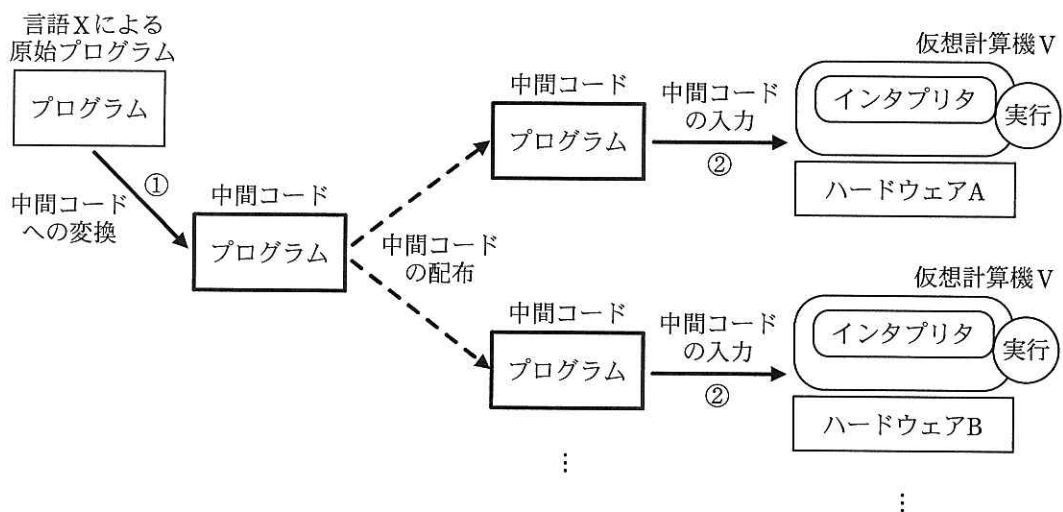


図 1 仮想計算機によるインタプリタ方式の実行例

〔図 1 の説明〕

- (1) 言語 X で記述された原始プログラムを、仮想計算機 V で実行できる中間コードに変換する (図 1①)。
- (2) 中間コードに変換されたプログラムを仮想計算機 V に入力し、V のインタプリタで逐次解釈しながら実行する (図 1②)。

aに関する解答群

- ア 特定の OS やハードウェアに依存しない
- イ 特定の OS やハードウェアに依存する
- ウ 特定のハードウェアの性能を引き出す
- エ 特定のハードウェアのメモリ使用量を低減する

設問 3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

仮想計算機 V におけるインタプリタ方式による実行時に、繰り返し実行される処理などの情報（以下、プロファイル情報という）を収集しておき、その内容を解析して、特定の処理の中間コードをプログラムの実行途中で実行形式プログラムに変換し、以後の実行に実行形式プログラムを利用する方法として、図 2 のような動的コンパイル方式がある。実行形式プログラムを実行途中から利用することによって、以後のプログラムの実行性能の向上が期待できるので、Java などの言語処理系で採用されている方式である。

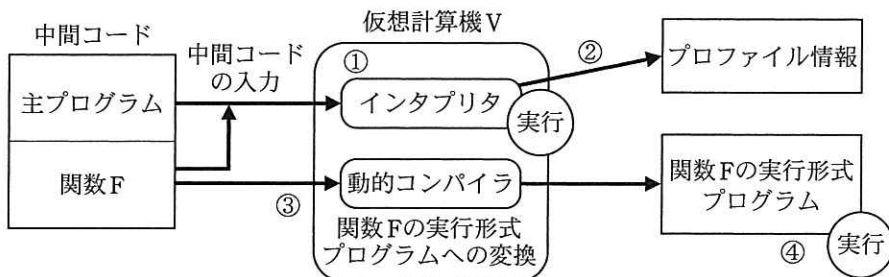


図 2 動的コンパイル方式を適用した処理の流れ

〔図 2 の説明〕

- (1) 主プログラムと主プログラムから呼び出される関数 F が中間コードに変換されたプログラムがある。
- (2) 仮想計算機 V のインタプリタは、中間コードを逐次解釈しながら実行する（図 2 ①）。このとき、関数 F が呼び出される回数をプロファイル情報として収集する（図 2 ②）。
- (3) 仮想計算機 V は、関数 F が呼び出される都度、プロファイル情報を解析し、関数 F を実行形式プログラムに変換するかどうかを判定する。

- (4) (3)で変換すると判定した場合、仮想計算機 V は、動的コンパイラを起動して関数 F を実行形式プログラムに変換する (図 2③)。
- (5) (4)で関数 F を変換した後は、関数 F が呼び出されたときには実行形式プログラムが実行される (図 2④)。

動的コンパイル方式を適用する場合には、実行時間に加えて動的コンパイラの起動時間やコンパイル時間を考慮する必要がある。図 2 に示す関数 F を、インタプリタ方式で実行した場合と、動的コンパイル方式で実行した場合の実行時間について考える。ここで、各方式の実行条件は、次の条件に示すとおりである。主プログラムから関数 F の呼出し回数が 400 回のとき、インタプリタ方式の場合の実行時間は 秒であり、動的コンパイル方式を適用した場合の実行時間は 秒である。

[条件]

- ・主プログラムの実行時間は考えない。
- ・プロファイル情報を収集する時間と、仮想計算機 V から実行形式プログラムを呼び出すのに必要な処理時間は考えない。
- ・関数 F の中間コードは 400 命令から成り、関数 F が 1 回呼び出されたときに実行する中間コードの命令数は、2,000 命令である。
- ・動的コンパイル方式を適用した場合、関数 F が 101 回目に呼び出されるときに動的コンパイラが起動され、関数 F の中間コードを実行形式プログラムに変換する。
- ・動的コンパイラの起動時間とコンパイル時間は、実行時間に含める。
- ・動的コンパイラの起動時間は 0.1 秒とし、コンパイル時間は、中間コード 1,000 命令当たりで 0.1 秒とする。
- ・インタプリタによる中間コード 1 命令の実行時間は 500 ナノ秒とし、中間コード 1 命令に対応する実行形式プログラムの実行時間は 10 ナノ秒とする。

b, c に関する解答群

- | | | |
|---------|---------|---------|
| ア 0.106 | イ 0.146 | ウ 0.206 |
| エ 0.246 | オ 0.4 | カ 0.406 |