

# ПРИЛОЖЕНИЯ

## ПРИЛОЖЕНИЕ 1

### ФОРМЫ ВЫХОДНЫХ ДОКУМЕНТОВ СИСТЕМЫ ПРОГРАММ ДЛПАН

Форма 6.1, лист 2

Производственная программа в натуральном и стоимостном выражении, тыс. руб.

Критерий: максимум 600 прибыль

ЭММ: план завода на III квартал

Код продукта	Наименование продукта	Единица измерения	Программа в натуральном выражении			Стоимость продукта в ценах критерия
			Минимум	Максимум	План	
1	2	3	4	5	6	7

Форма 6.2, лист 4

Расчет показателей и потребности ресурсов на программу, тыс. руб.

Критерий: максимум 604 прибыль

ЭММ: план завода на III квартал

Код показателя (ресурса)	Наименование показателя (ресурса)	Единица измерения	Вид ограничения	Фонд показателя (ресурса)	Необходимо на программу	(гр. 6: гр. 5) × 100%	Двойственный оценщик
1	2	3	4	5	6	7	8

Форма 6.3, лист 1

Расчет показателей критериев, тыс. руб.

Критерий: максимум 600 прибыль

ЭММ: план завода на III квартал

Код показателя	Наименование показателя	Единица измерения	Вид критерия	Значение критерия
1	2	3	4	5

План производства на год по заводу, тыс. руб.

Критерий: максимум 600 прибыль

ЭММ: план завода на III квартал

Код продукта	Наименование продукта	Единица измерения	Годовой план		В том числе по кварталам							
			Количество	Стоимость	I		II		III		IV	
					Количество	Стоимость	Количество	Стоимость	Количество	Стоимость	Количество	Стоимость
1	2	3	4	5	6	7	8	9	10	11	12	13

Форма 6.5, лист 1

План производства на год по цеху, тыс. руб.

Критерий: максимум 600 прибыль

ЭММ: план завода на III квартал

Код продукта	Наименование продукта	Единица измерения	Годовой план		В том числе по кварталам							
			Количество	Стоимость	I		II		III		IV	
					Количество	Стоимость	Количество	Стоимость	Количество	Стоимость	Количество	Стоимость
1	2	3	4	5	6	7	8	9	10	11	12	13

Форма 6.6, лист 1

План загрузки цехов основного производства на год, тыс. руб.

Критерий: максимум 600 прибыль

ЭММ: план завода на III квартал

Номер цеха	Загрузка цехов по кварталам											
	I			II			III			IV		
	Полный фонд времени	Необходимый фонд времени	% использования фонда времени	Полный фонд времени	Необходимый фонд времени	% использования фонда времени	Полный фонд времени	Необходимый фонд времени	% использования фонда времени	Полный фонд времени	Необходимый фонд времени	% использования фонда времени
1	2	3	4	5	6	7	8	9	10	11	12	13

План производства на квартал по цеху, тыс. руб.

Критерий: максимум 600 прибылей  
ЭММ: план завода на III квартал

Наименование продукта	Наименование цеха	Квартал	План на квартал		В том числе по месяцам					
			Количество	Стоимость	1-й месяц		2-й месяц		3-й месяц	
					Количество	Стоимость	Количество	Стоимость	Количество	Стоимость
1	2	3	4	5	6	7	8	9	10	11

План производства на квартал по заводу, тыс. руб.

Критерий: максимум 600 прибылей  
ЭММ: план завода на III квартал

Наименование продукта	Наименование цеха	Квартал	План на квартал		В том числе по месяцам					
			Количество	Стоимость	1-й месяц		2-й месяц		3-й месяц	
					Количество	Стоимость	Количество	Стоимость	Количество	Стоимость
1	2	3	4	5	6	7	8	9	10	11

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
МАШИНОЙ ГРАФИКИ В ТЕВУС<sup>1</sup>

Наиболее распространенным видом графического программного обеспечения являются графические системы, ориентированные на периферийную многопользовательскую конфигурацию, реализованную в терминальной вузовской системе ТЕВУС.

Особенность терминальной ЭВМ, входящей в состав рабочего места пользователя графического прикладного программного обеспечения, заключается в ограниченности вычислительных, информационных и других видов ресурсов. Возможности графических программ реализуются с помощью цветного графического растрового дисплея, входящего в состав терминального места.

Периферийные устройства ввода-вывода центральной вычислительной машины восполняют недостаток, например, информационных ресурсов, которые необходимы пользователю терминальной ЭВМ для решения его задачи.

Достоинства распределенных графических систем по сравнению с вычислительными системами с протестями, без микропроцессоров, графическими терминалами следующие<sup>2</sup>:

1. Экономичность. Благодаря передаче терминальной ЭВМ всех функций управления графической программой и оборудованием существенно уменьшается объем данных, проходящий по линиям дистанционной связи и состоящий теперь только из информации, требуемой для нормального счета графической программы. Центральная ЭВМ освобождается для работы, действительно требующей ее мощных ресурсов.

2. Большая скорость реакции системы. Выполнение графической программы на терминальном рабочем месте позволяет отвечать на интерактивные запросы пользователя без применения центральной ЭВМ и линий связи, что обеспечивает быстрый отклик графической системы.

3. Сервисное обеспечение. Задача написания графического программного обеспечения ТЕВУС решена путем создания пакета подпрограмм, к которым обращаются из прикладной программы, написанной на языке высокого уровня ФОРТРАН-IV. Подпрограммы машинной графики служат для обеспечения графического взаимодействия пользователя с ЭВМ. В зависимости от степени предоставления сервисных возможностей графические подпрограммы могут быть подразделены на два основных уровня — базовые подпрограммы и прикладные подпрограммы.

В ИГУ разработаны следующие базовые подпрограммы:

1) быстрое закрасивание экрана нужным цветом — CALL ERASE (C), где C = 0 означает черный цвет, C = 1 — красный, C = 2 — синий, C = 3 — зеленый);

2) выставляющие точки цвета C в позицию с координатами Y и X (Y и X изменяются от 0 до 255) — CALL POINT (C, X, Y);

3) проведение вектора цветом C на начала координат в точку с координатами X и Y — CALL VECTOR (C, X, Y);

4) вывод символа "А" цветом C на экран — CALL DCHAR (C, "A").

При помощи этих базовых подпрограмм Ю. Я. Сальников разработал набор подпрограмм для графического представления аномической информации в виде гистограмм различного цвета и расположения. По желанию пользователя на одном экране может быть представлено несколько графиков на соответствующих мини-экранах. При обращении к подобным графическим подпро-

<sup>1</sup> Написано в соавторстве с Ю. Я. Сальниковым.

<sup>2</sup> См.: Бобков В. А., Белов С. Б., Соловьев И. А. Распределенные графические системы. — В кн.: Машинная графика баз данных. — М.: ИД МЦНТИ, 1984. — Вып. 26. — С. 66—67.

граммам либо в режиме диалога, либо в заранее фиксированном виде при помощи подпрограмм типа XXXGRA задаются следующие параметры:

KLF — «ключ» для задания цвета экрана (—1— предыдущий экран не затрывается; 0 — затрывается черным цветом; 1 — затрывается красным цветом; 2 — затрывается синим цветом; 3 — затрывается зеленым цветом);

KLRAM — «ключ» определяет цвет рамки и контуров гистограммы аналогично KLF;

KLG — «ключ» для определения цвета гистограммы. Может принимать значения от 0 до 15. Каждому значению соответствует определенный цвет, образуемый сочетанием четырех базовых цветов в квадрате из четырех точек  $2 \times 2$ ;

KGX — количество мини-экранов для вывода графиков на большом экране ( $256 \times 256$  точек) по оси X;

KGY — количество мини-экранов по оси Y;

NGX — номер мини-экрана по оси X, отсчет от левого нижнего угла;

NGY — номер мини-экрана по оси Y, отсчет от левого нижнего угла;

NG — номер мини-экрана, выводится цветом рамки в правом верхнем углу мини-экрана.

Для построения обычных гистограмм можно применить подпрограмму SCR.

Обращение к ней:

CALL SCR (KEY, KLF, KLRAM, KLG, KGX, NGX, KGY, NGY, NG, KZ, SM, ZN),

где KZ — число элементов в массиве; ZN(KZ) — массив значений элементов (например, RC(KP)); SM — максимальное значение, которое либо задается явно, либо определяется в подпрограмме SCR; KEY — «ключ» для задания вида графика.

Если KEY = 1, то прямоугольные гистограммы не закрашиваются и высота  $H$  прямоугольника  $H$  (их число по оси абсцисс равно KZ) пропорциональна

$ZN(I); \sum_{I=1}^{KZ} ZN(I) + 100$ . Если KEY = 2, то  $H(I)$  пропорциональна  $ZN(I);$

$(\max_{I=1, KZ} ZN(I))$ . При KEY = 3 и KEY = 4 на экране вычерчиваются

гистограммы, соответствующие двум предыдущим видам, но окрашенные в цвет, определяемый KLG.

Подпрограмма SCRKL аналогична SCR, но дополнительно задается массив KLN(KZ) (он идет последним в списке формальных параметров). Элементами этого массива являются числа от 0 до 15, в соответствии с которыми будет задан цвет прямоугольничков гистограммы.

Подпрограмма SCRSR имеет те же формальные параметры, что и SCR, но перед ZN стоит ZS. Это массив размерности KZ, значения которого соответствуют некоторым плавным (базовым, лимитирующим) показателям. С его помощью осуществляется наложение двух гистограмм для массивов ZS и ZN.

При этом наименьший прямоугольничок рисуется синим цветом, а превышение — красным (если  $KEY = 1$  V2 и  $ZS(I) > ZN(I)$ ) или зеленым цветом (если  $KEY = 1$  V2 и  $ZS(I) < ZN(I)$ ).

Подпрограмма SCSRKL объединяет возможности двух предыдущих подпрограмм: наименьшее значение выводится цветом KLN2, а отклонение ZS от ZN окрашивается красным или зеленым цветом.

Разработаны и другие подпрограммы для лучшего восприятия экономической информации. Например, каждый прямоугольничок гистограммы, отображающей себестоимость единицы продукции, разбивается на несколько частей, окрашенных в разные цвета, соответственно значениям элементов затрат по статьям калькуляции. В этом случае массив ZN задается в виде матрицы.

## ПРИЛОЖЕНИЕ 3

### РАСШИРЕНИЕ АДАНТИВНЫХ ВОЗМОЖНОСТЕЙ ДИАЛОГОВОГО КОМПЛЕКСА ТЕХНИКО-ЭКОНОМИЧЕСКОГО ПЛАНИРОВАНИЯ ДЛЯ МАЛЫХ И СРЕДНИХ ПРЕДПРИЯТИЙ<sup>1</sup>

Работы по созданию и внедрению диалогового комплекса технико-экономического планирования для малых и средних предприятий показали, что используемый в качестве базового алгоритмический язык ФОРТРАН недостаточно эффективен при разработке адантивного, легко модифицируемого программного обеспечения. Этот факт оказался особенно оптимистичным при работе с мини-ЭВМ. Поэтому для расширения свойств адантивности был использован универсальный язык программирования — С. Не останавливаясь на его описании и принципах работы, отметим некоторые особенности нового варианта программного обеспечения диалогового комплекса на основе языка С.

В процессе написания, отладки и эксплуатации программ особое место занимает организация ввода-вывода. Первоначально для этих целей применялись файлы последовательного доступа к данным, размещенным в соответствии с определенными форматами. Чтение и запись данных происходили по этим форматам.

Ошибка при корректровке файла или необходимость изменения структуры данных вызывали собой в работе и требовали вмешательства специалиста. Использование же языка С позволило отказаться от строгого соответствия данных определенному формату. Границами поля в этом случае являются пробелы. Например, ввод строки

ТОВАР 11.500

на языке ФОРТРАН требует такой последовательности операторов:

```
READ (1, 1) PEN1, PEN2, TOV
```

```
1 FORMAT (2A4, F7.3)
```

на языке С для аналогичной операции достаточно записать:

```
fscanf (fp, "%s%f", pen, &to);
```

При этом появление новых пробелов или корректровка данных не приведут к ошибке при счете.

Имя файла ввода-вывода можно задавать в режиме диалога или читать из файла справочника. Таким образом, файл в программе на языке С можно открыть следующей операцией:

```
if (fp=fopen (name-file, mode)) == NULL {  
    printf ("НЕ МОГУ ОТКРЫТЬ ФАЙЛ %s\n",  
    name-file);};
```

где строка name-file содержит имя файла, mode — символьная переменная, которая задает режим (чтение, запись, дополнение).

<sup>1</sup> Приложение подготовлено А. А. Перфильевым.

<sup>2</sup> См., например: Каринган Б., Риччи Д., Фьюэр А. Язык программирования Си: Задачи по языку Си. — М.: Финансы и статистика, 1983.

Работа с большими массивами данных требует резервирования памяти. Это делается при помощи оператора DIMENSION (в ФОРТРАНе). В процессе счета не все массивы используются одновременно, поэтому они являются своего рода «балластом».

В С возможно временное резервирование памяти под некоторое количество объектов (данных). Когда работа с зарезервированной памятью закончена, ее можно освободить и затем затребовать в других частях программы. Например, нам потребовался массив для размещения 100 чисел целого типа. В этом случае обращаем

```
buffer=calloc(100, sizeof(int));
```

мы резервируем функцией calloc память, достаточную для 100 чисел целого типа, и передаем начальный адрес полученного массива в переменную buffer.

Обращение free (buffer) освобождает память, на которую указывала переменная buffer.

Ограничения на располагаемую память привели к тому, что часть данных в процессе счета находится на МД и вызывается в оперативную память по мере необходимости. В этом случае частота обращений к нужной информации и возникает необходимость прямого обращения к нужной информации в файле. Поэтому файл основного производства был несколько изменен.

Все записи файла (каждая запись представляет нормативно-технологическую карту (НТК)) разбиты на подзаписи. Каждая подзапись помечена указателем:

- 1#K — указатель начала НТК и первой подзаписи, которая включает название изделия, цены оптовую и розничную, НЧП, зарплату и др.;
- 2#A — указатель принадлежности к ассортиментным группам;
- 3#M — указатель на подзапись норм расхода материалов;
- 4#T — указатель на подзапись норм трудоемкости по основным профессиям.

При запуске программы файл основного производства открывается, функция шифре просматривает файл до указателя и возвращает адрес его местоположения в файле. Адрес запоминается и в дальнейшем используется при поиске требуемой информации.

Отсутствие той или иной подзаписи не влечет прекращения работы, не будут функционировать лишь те блоки, в которых опущенная информация задействована непосредственно.

Стандартная форма вывода на экран дисплея мини-ЭВМ предполагает, что каждая новая строка вымывает сдвиг старой на одну строку вверх.

Практические занятия со студентами и специалистами промышленных предприятий показали, что более приемлемая форма, при которой данные помещаются в определенные фрагменты экрана и при изменениях меняют свои значения, оставаясь на месте. Библиотека экранных функций языка С позволяет реализовать эти возможности. Можно назвать несколько таких функций:

- 1) clrscr — очистка экрана,
- 2) clrnl — очистка строки,
- 3) scout — вывод информации с координатами строки, столбца и длиной вывода.

В пояснение можно привести следующую программу: для clrscr (1, 1), например, производится стирание экрана с 1-й строки и 1-го столбца,

a scout (10, 5, "ДИАЛОГОВЫЙ КОМПЛЕКС ТЕХНИКО-ЭКОНОМИЧЕСКОГО ПЛАНИРОВАНИЯ") выводит текст в навывчках на экран с 10-й строки и 5-го столбца. Для вывода данных удобна следующая запись: printf (str, "ОБЩИЙ ВЫПУСК %d ЕДИНИЦ", v); при помощи ее в строку str передается текст из поля формата, расположенного между навывчками. Символ % означает спецификацию формата данных; вместо %d в str будет выдано значение переменной v (v — переменная целого типа). scout (12, 1, str); — сформированная таким образом строка выводится на экран.

Размер оперативной памяти в программах на языке С удается сократить и за счет передачи в подпрограммы (функции) не самих объектов, а их адресов.

Например, и функцию reverse

```
reverse (&a[0], &b[0]);
```

Передается адреса двух массивов a и b.

Использование С позволяет непосредственно подключать блоки программы, написанные на языке АССЕМБЛЕР и ФОРТРАП, обращаясь к ним, как к подпрограммам. Если обмен данных происходит из задач (загруженных модулей), то он может осуществляться через файлы. При этом запуск задач можно осуществлять при помощи библиотечных функций языка С.

Созданный вариант диалогового комплекса технико-экономического планирования позволяет практически без изменений использовать один и те же задачи для расчетов годовых планов различных предприятий, настраиваясь на исходной информации.