

টেক্সট মোডে, টেক্সট অ্যাট্রিবিউট বাইটের সাহায্যে স্ক্রীন নিয়ন্ত্রণ

আবদুল্লাহ আল সালাম আহমেদ

(পূর্ব প্রকাশিতের পর)

সফটওয়্যারের সাহায্যে অ্যাট্রিবিউট বাইট নিয়ন্ত্রণ ও এর প্রয়োগ :

একদে পুটি ব্যালক প্রদর্শিত ভাষা Pascal ও C এর সাহায্যে অ্যাট্রিবিউট বাইট নিয়ন্ত্রণ ও এর প্রয়োগ নিয়ে আলোকপাত করা যাক।

প্রারম্ভিক করণীয় কার্যক্রম :

প্রোগ্রামের শুরুতে Pascal 5.5 এ 'cr1' ইউনিকোড ব্যবহার করতে হবে এবং Turbo C তে 'CONIO.H' এই হেডার ফাইলটিকে অন্তর্ভুক্ত (include) করতে হবে।

নিম্নের আদোচায়ের আমরা প্রথমে টেক্সট অ্যাট্রিবিউট নিয়ন্ত্রণের জন্য Pascal ও C তে প্রদত্ত কিছু সুযোগ (facility) নিয়ে আলোচনা করবো এবং শেষে সরাসরি অ্যাট্রিবিউট বাইট নিয়ন্ত্রণ করার প্রক্রিয়াটি জানাবো।

টেক্সট অ্যাট্রিবিউটকে প্রোগ্রাম চলার দিক পূর্বাঙ্কুর আনার কৌশল :

প্রোগ্রামের অ্যাট্রিবিউটের যে কোন অবস্থা, টেক্সট অ্যাট্রিবিউটের অবস্থা ঘাই থাক বা কোন প্রোগ্রামটির ঠিক তরুর অবস্থায় টেক্সট অ্যাট্রিবিউটের যে অবস্থা ছিল সে অবস্থা ফিরিয়ে আনতে হল Pascal-এ Normvideo; লিখতে হবে। C তে লিখতে হবে Normvideo (); (এখানে উল্লেখ্য যে টেক্সট অ্যাট্রিবিউটকে একবার কোন মানে সুচিত (define) করলে পরবর্তীতে টেক্সট অ্যাট্রিবিউটকে অন্য কোন মানে সংজ্ঞায়িত (define) করার পূর্ব পর্যন্ত সমস্ত টেক্সট অক্ষর/চিহ্ন পূর্বেক অ্যাট্রিবিউটে চিহ্নিত হবে।)

উদাহরণ দিয়ে Normvideo ব্যাপারটি ব্যাখ্যা করা যাক। ধরা যাক কোন প্রোগ্রাম (শুরু করার আগে টেক্সট অ্যাট্রিবিউট স্বাভাবিক ছিল। অর্থাৎ টেক্সট কালার ছিল সাদা এবং টেক্সট ব্যাকগ্রাউন্ড ছিল কালো। এখন প্রোগ্রামটিতে কোন এক সময় টেক্সট অ্যাট্রিবিউট বাইটকে বিপরীত ভিত্তানে (Reverse video) সুচিত (define) করে ঘনিয়ে কিছু লেখা হলো। এখন এর পর থেকে ঘনিয়ে টেক্সট অ্যাট্রিবিউটকে প্রোগ্রামের ঠিক তরুর অবস্থায় ফেরান ছিল আনতে হলো প্রোগ্রামে normvideo(Pascal-> normvideo ; C->normvideo (); লিখতে হবে। এখন ঘনিয়ে যে টেক্সট চিহ্নায়িত করা য়োক বা কোন ডা স্বাভাবিক (টেক্সট কালার সাদা, টেক্সট ব্যাকগ্রাউন্ড কালো) অবস্থায় চিহ্নিত হবে। ঘনিয়ে পূর্বলিখিত বিপরীত ভিত্তানের টেক্সট কিছু অধিকৃত থাকবে। পুনরায় যুখে ফেরান ঘনিয়ে প্রোগ্রামের শুরুত অবস্থায় আনতে হলো normvideo বোনার পর স্ক্রীন ক্লিয়ার করতে হবে (Pascal এ -> clrscr ; turbo C এ -> clrscr ();)

উচ্ছ্বাস বিট (বিট(০)) নিয়ন্ত্রণ :

এই বিট '০' লিখে হল Pascal এ লিখতে হবে Lowvideo; C তে লিখতে হবে lowvideo (); বিটটিতে '১' লিখে সুচিত (define) করতে হলো (অর্থাৎ বিটটি 'অনু' করতে হবে) Pascal এ লিখতে হবে Highvideo; C তে সেক্ষেত্রে লিখতে হবে highvideo (); এই বিটটি নিয়ন্ত্রণ করে কোন টেক্সট পদ্য লেখলে, এই বিটের শেষ মান অনুযায়ী পর্যায় টেক্সটটি অনুচ্ছ্বাস বা উচ্ছ্বাস ভাবে প্রদর্শিত হবে।

অক্ষর / চিহ্নের বর্ণ নিয়ন্ত্রণ

Pascal/C তে কোন টেক্সটকে কোন রং-এ (কালার ঘনিয়ে ফেরানোর ক্ষেত্রে প্রযোজ্য) চিহ্নায়িত করতে হলো উক্ত টেক্সট ঘনিয়ে লেখতে পাঠানোর পূর্বে লিখতে হবে textcolor (color name অথবা color decimal value অথবা color hexadecimal value) নিম্নের তালিকায় এটির ব্যাখ্যা দেয়া হলো।

জাভা	রং এর নাম	ডেসিমেল	হেক্সা ডেসিমেল সংখ্যা
PASCAL	তালিকা-১ এর ১৬টির যে কোনটির নাম	0.15 (যে কোনটি)	\$00.\$0F (যে কোনটি)
C	ঐ	ঐ	0X00.0X0F (যে কোনটি)

উদাহরণ : যেমন টেক্সটকে হলুদ বর্ণ সুচিত (define) করতে হলো টেক্সট লেখার পূর্ব Pascal এ লিখতে হবে textcolor (Yellow) বা textcolor (14) বা textcolor (0XE)

C তে একইভাবে হলুদ বর্ণ সুচিত (define) করার ক্ষেত্রে textcolor (Yellow) বা textcolor (14) ; বা textcolor (0XE) লিখে তারপর টেক্সট পদ্য লিখতে হবে। অর্থাৎ C তে 'Computer Jagat' কথাটি হলুদ বর্ণ লিখতে হলো নিম্নের পুটি লাইন লিখতে হবে।

```
textcolor (Yellow);
cprint ("Computer Jagat");
Pascal এ পূর্বেক কথাটি লিখতে হলো নিম্নের পুটি লাইন লিখতে হবে :
textcolor (Yellow);
write ("Computer Jagat")
```

(যে প্র টেক্সট অ্যাট্রিবিউটের বিভিন্ন বৈশিষ্ট্য আনতে হলো টেক্সট পর্যায় লেখার ক্ষেত্রে Pascal 5.5 এ স্বাভাবিকভাবে write/writeln কাছ করলেও turbo C তে printf () কাছ করতে না। সেক্ষেত্রে cprint () নিয়ে টেক্সট লেখতে হবে।)

অক্ষর / চিহ্নের দৃশ্য-বিনোদ (Blink) নিয়ন্ত্রণ

Pascal/C তে textcolor এর মতো প্যারামিটার (Parameter) হিসেবে কোন রং-এর নাম/সংখ্যা লিখে +Blink বা +128 বা +\$80 (Pascal-এ) & + 0 x 80 (C-তে) লিখলে তারপর টেক্সট পর্যায় লিখলে সেটি দৃশ্য হয়।

যেমন : সন্ধ্য বর্ণের কোন টেক্সট দৃশ্য করতে হলো প্রথম লিখতে হবে :

Pascal	C
textcolor (Green + Blink)	textcolor (Green + Blink)
বা	বা
textcolor (Green + 128)	textcolor (Green + 128)
বা	বা
textcolor (2 + \$80)	textcolor (2 + 0 x 80)

অতঃপর টেক্সটকে ঘনিয়ে পাঠাতে হবে।

অক্ষর / চিহ্নের পটভূমির রং (Background Color) নিয়ন্ত্রণ :

Pascal/C তে কোন টেক্সট এর পটভূমির রং (শুধু কালার ঘনিয়ে ফেরানোর ক্ষেত্রে প্রযোজ্য) কোন বর্ণ সুচিত (defined) করতে হলো প্রথমে লিখতে হবে text background (Color name অথবা color decimal no. অথবা color Hexadecimal no) তবে এ ক্ষেত্রে তালিকা-১ এর মতো সংখ্যা (যে কোন সংখ্যা) কেবল প্রথম ৮টি রং ব্যবহার করা যাবে।

জাভা	রং এর নাম	ডেসিমেল সংখ্যা	হেক্সা ডেসিমেল সংখ্যা
Pascal	১-৯ তালিকার প্রথম ৮টির যে কোনটির নাম	0.7 (যে কোনটি)	\$00.\$07 (যে কোনটি)
C	"	0.7 (")	0x00.0x07 (")

তালিকা ও টেক্সট এর ব্যাকগ্রাউন্ড কালার নিয়ন্ত্রণ

উদাহরণ : যেমন কোন টেক্সট এর ব্যাকগ্রাউন্ড এর বর্ণ সন্ধ্য নির্ধারণ করতে হলো প্রথমে লিখতে হবে :

Pascal	C
textbackground (GREEN)	textbackground (GREEN)
বা	বা
textbackground (2)	textbackground (2)
বা	বা
textbackground (\$02)	textbackground (0 x 02)

এরপর যে টেক্সটই লেখা হোক না কেন, তার ব্যাকগ্রাউন্ডের বর্ণ হবে সন্ধ্য।

সরাসরি অ্যাট্রিবিউট বাইট নিয়ন্ত্রণ :

Pascal/C সুই ভাষাতেই সরাসরি অ্যাট্রিবিউট বাইট নিয়ন্ত্রণ করা যায়। এতে সুবিধা হলো অ্যাট্রিবিউট বাইটের যে কোন বিট ইচ্ছতে অন/অফ (১/০) করে অ্যাট্রিবিউট বাইটকে যে কোন কন্ডিশনে সংজ্ঞায়িত (defined) করা যায় তথা টেক্সট অ্যাট্রিবিউটকে সম্পূর্ণ নিয়ন্ত্রণ রাখা যায়।

Pascal-এ সরাসরি আর্দ্রবিট বাইট নিয়ন্ত্রণ :

এছাড়া যে কপিবেশন আনতে হবে তার হেডারভিত্তিক সংখ্যা বা ডেসিমালা সংখ্যা textatr এর নিচে (assign) হয়। Pascal এর crt ইউনিট textatr ব্যারিবেল (variable) টি সংজ্ঞায়িত (define) করা আছে এ কারণে এটিকে প্রোগ্রামে সূচিত (declare) করার প্রয়োজন নেই। উদাহরণ স্বরূপ; হনোক্রেম মনিটরে কোন টেক্সটে আ্যাপলাইন আনান হলো টেক্সটটি বক্সিং লিখতে পাঠানোর পূর্বে টেক্সট আর্দ্রবিট বাইটটিকে, নিম্নোক্ত ক্যাটাগি লিখে সংজ্ঞায়িত করতে হয় :

```
textatr := $01;           0 0 0 0 0 0 0 1
    বা
    (টেক্সট আর্দ্রবিট বাইট)
```

```
textatr := 1;
```

যদি আ্যাপলাইন ও ব্লিঙ্ক এ দুই বৈশিষ্ট্য টেক্সটে একই সাথে আনতে হয় তবে লিখতে হবে

```
textatr := $81;           1 0 0 0 0 0 0 0 1
    বা
    (টেক্সট আর্দ্রবিট বাইট)
```

```
textatr := 129;
```

কালার মনিটরের জন্য উপরোক্ত নির্দেশগুলি মনিটরে যথাক্রমে কালো ব্যাকগ্রাউন্ডে নীল টেক্সট এবং কালো ব্যাকগ্রাউন্ডে ব্লিঙ্ক অবস্থায় নীল টেক্সট প্রিয়ারিত করবে।

Turbo C -এ সরাসরি আর্দ্রবিট বাইট নিয়ন্ত্রণ

C ভাষায় conio.h হেডার ফাইলে textatr নামে ভয়েড (void) সঙ্গায়িত (define) করা আছে বা প্যারামিটার হিসেবে একটি বাইট সংখ্যা (Byte value) গ্রহণ করে এবং টেক্সট এর বর্ণমালা আর্দ্রবিট, ই যানে সূচিত (define) করে। সত্যরূপে উদাহরণ স্বরূপ হনোক্রেম মনিটরে টেক্সটে আ্যাপলাইন আনতে হলে টেক্সট মনিটরে লিখতে পাঠানোর পূর্বে টেক্সট আর্দ্রবিটটিকে নিম্নোক্ত ক্যাটাগি সংজ্ঞায়িত করতে হয়।

```
textatr (0 x 01);       0 0 0 0 0 0 0 0 1
    বা
    (টেক্সট আর্দ্রবিট বাইট)
```

```
textatr (1);
```

যদি আ্যাপলাইন ও ব্লিঙ্ক এ দুই বৈশিষ্ট্য টেক্সটে একই সাথে আনতে হয় তবে লিখতে হবে

```
textatr (0 x 81);       1 0 0 0 0 0 0 0 1
    বা
    (টেক্সট আর্দ্রবিট বাইট)
```

```
textatr (129);
```

কালার মনিটরে জন্য উপরোক্ত নির্দেশগুলি মনিটরে যথাক্রমে কালো ব্যাকগ্রাউন্ডে নীল টেক্সট এবং কালো ব্যাকগ্রাউন্ডে ব্লিঙ্ক অবস্থায় নীল টেক্সট প্রিয়ারিত করবে।

প্রোগ্রামটি শেষ হওয়ার পর যাক মনিটরটি, প্রোগ্রামটি শুরু হওয়ার আগের আর্দ্রবিট সূচিত (initialize) হয় সে কারণে প্রোগ্রামের শেষে normvideo করে স্ক্রীন পরিষ্কার (clear) করে নিতে হবে। একে মনিটরটি প্রোগ্রাম শুরু হওয়ার পূর্বে অবস্থায় সূচিত (initialize) হবে। এটি না করলে প্রোগ্রামের টেক্সট আর্দ্রবিটের কোন খান প্রোগ্রামটি শেষ হওয়ার পর মনিটরে আনতিমোক্ত আর্দ্রবিট সংজ্ঞায়িত (define) করে রাখতে পারে।

নিকট দুটি প্রোগ্রাম দেয়া হলো। প্রোগ্রামের একই কাছ করে। একটি Pascal 5.5 এ অপরটি turbo C তে। প্রোগ্রামের হনোক্রেম মনিটরে জন্য দশাঙ্ক স্বরূপ স্বয়ং আর্দ্রবিট কপিবেশন দেয়া হয়েছে। এছাড়া আর্দ্রবিট বাইটিক সরাসরি নিয়ন্ত্রণ করে প্রোগ্রাম লেখা হয়েছে। যেহেতু আর্দ্রবিট বাইটের সাহায্যে টেক্সটের ও এর পটভূমির (Background) বর্ণ নিয়ন্ত্রণের সম্ভব বিস্তারিত আলোচনা পূর্বে করা হয়েছে তাই কালার মনিটরে ব্যবহারকারীর ও টেক্সটের ও এর পটভূমির বিভিন্ন ইং-এর বৈচিত্র্য আনতে আশা করা বৈশিষ্ট্য দেয়া হলো।

বিঃ দ্রঃ অনেকক্ষেত্রে সারণ্য চক্রের অনুচ্ছল (low intensity) ও উচ্ছল (high intensity) এর মধ্যে পার্থক্য লোখা যায় না। সেছাড়া মনিটরের intensity knob দুইয়ে এ দুটির পার্থক্য বুঝতে পারেন।

/* This is a pascal program Which demonstrate various text of different combinations in monochrome monitor. */

```
uses crt;
Begin
  clrscr;
  writeln('Computer Jagat < normal >');
  writeln;
  textatr := $01;
```

```
writeln('Computer Jagat <Normal (intensed)>');
  writeln;
  textatr := $87;
  writeln('Computer Jagat <Blinking/without Intensed>');
  writeln;
```

```
textatr := $81;
  writeln('Computer Jagat <Blinking(intensed)>');
  writeln;
```

```
textatr := $70;
  writeln('Computer Jagat < Reversed Video>');
  writeln;
```

```
textatr := $10;
  writeln('Computer Jagat <Reversed Vide (blinking)>');
  writeln;
```

```
textatr := $01;
  writeln('Computer Jagat < Underline (Without Intensed)>');
  writeln;
```

```
textatr := $09;
  writeln('Computer Jagat <Under Line (Intensed)>');
  writeln;
```

```
textatr := $89;
  writeln('Computer Jagat <Under line (Blinking & Intensed)>');
  writeln;
```

```
textatr := $00;
  writeln('<Hidden line-Textcolor(black) textbackground(black)>');
  writeln;
```

```
repeat until keypressed;
  normvideo;
  clrscr;
  end.
```

/* This is a Turbo C program Which demonstrate various test of different combinations in monochrome monitor. */

```
#include <conio.h>
main()
{
  clrscr();
  printf("\nComputer Jagat < normal >");
  textatr(0x01);
  printf("\nComputer Jagat <Normal (intensed)>");
  textatr(0x87);
  printf("\nComputer Jagat <Blinking (without intensed)>");
  textatr(0x81);
  printf("\nComputer Jagat <Blinking (intensed)>");
  textatr(0x70);
  printf("\nComputer Jagat < Reversed Video>");
  textatr(0x10);
  printf("\nComputer Jagat <Reversed Video (blinking)>");
  textatr(0x01);
  printf("\nComputer Jagat <YUnderline (Without Intensed)>");
  textatr(0x09);
  printf("\nComputer Jagat <Under Line (intensed)>");
  textatr(0x89);
  printf("\nComputer Jagat <Under line (Blinking & Intensed)>");
  textatr(0x00);
  printf("\n<Hidden line-Textcolor(black) textbackground(black)>");
  getch();
  normvideo();
  clrscr();
}
```

সমাপ্ত

●● অবস্থানটি আল সালেহ আহামেদ হাকীমুল হিন্দুবিদ্যালয়ের কম্পিউটার সার্ভিস এন্ড ইঞ্জিনিয়ারিং বিভাগের শেষ বর্ষের ছাত্র।