

به نام خدا



طراحی زبان‌های برنامه‌سازی

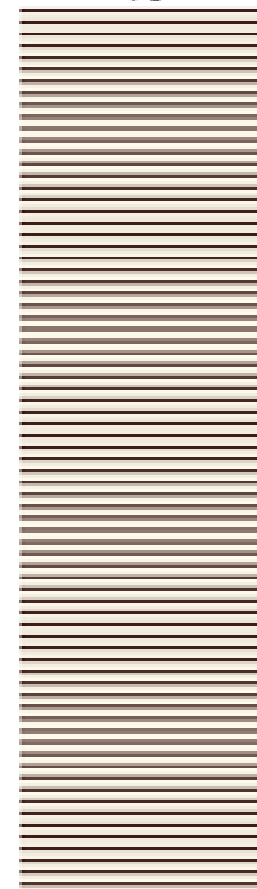
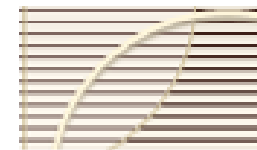
مرجان عظیمی
Azimi.marjan@gmail.com

فصل دهم

مدیریت حافظه

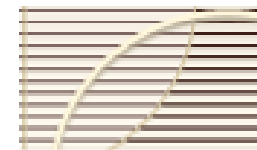
مدیریت حافظه

- با اینکه برنامه نویس با مدیریت حافظه سروکار دارد و باید برنامه هایی بنویسد که از حافظه به خوبی استفاده کند احتمالاً کنترل مستقیم او بر حافظه اندک است.

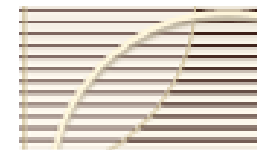


عناصری که در زمان اجرا به حافظه نیاز دارند

- سگمنت کد برای برنامه ترجمه شده کاربر
- برنامه های زمان اجرای سیستم
- ثوابت و ساختمان داده های تعریف شده توسط کاربر
- نقاط برگشت زیربرنامه ها
- محیطهای ارجاع
- حافظه های موقت در ارزیابی عبارات
- حافظه های موقت برای انتقال پارامترها
- بافرهای ورودی - خروجی
- عملیات فراخوانی و برگشت از زیربرنامه
- عملیات ایجاد و از بین بردن ساختمان داده ها
- عملیات درج و حذف اجزای ساختمان داده ها

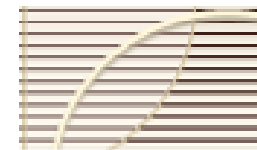


مدیریت حافظه تحت کنترل برنامه نویس و سیستم



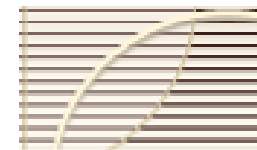
- مشکل کنترل برنامه نویس بر روی مدیریت حافظه:
- مسئولیت سنگینی را متوجه برنامه نویس می کند و ممکن است با مدیریت حافظه تحت کنترل سیستم تداخل ایجاد کند.
- مراحل مدیریت حافظه
 - تخصیص اولیه
 - بازیابی حافظه
 - فشرده سازی و استفاده مجدد

مدیریت حافظه ایستا



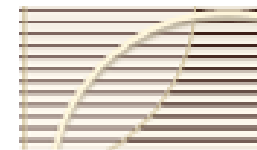
- ساده ترین شکل تخصیص، تخصیص ایستا است . این تخصیص در زمان ترجمه انجام می شود و در طول اجرا ثابت است.
- تخصیص حافظه ایستا کارآمد است زیرا در زمان اجرا هیچگونه زمانی برای مدیریت حافظه مورد انتظار نیست.

مدیریت حافظه هرم



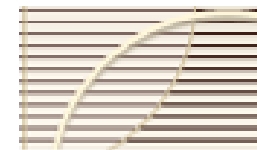
- هرم بلوکی از حافظه است که در آن قطعاتی از حافظه به روش غیرساخت یافته تخصیص می یابند و آزاد می شوند.
- تکنیکهای مدیریت حافظه هرم برحسب اینکه اندازه عناصر تخصیص یافته ثابت باشد یا متغیر به دو دسته تقسیم شوند.

مدیریت حافظه هرم (ادامه)



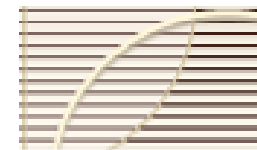
- مدیریت حافظه هرم با عناصر طول ثابت
- برای تخصیص یک عنصر اولین عنصر لیست فضای آزاد از لیست حذف می شود و اشاره گری به آن عنصر به عملیات درخواست کننده حافظه برگردانده می شود.
- اصلاح: شمارش ارجاعها و زباله روبی
 - برنامه نویس یا سیستم آنها را بر می گرداند.
 - شمارش ارجاع
 - زباله روبی
 - نشانه گذاری
 - جارو کردن

مدیریت حافظه هرم (ادامه)



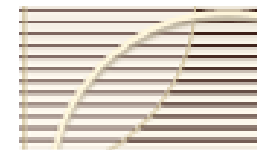
- مدیریت حافظه هرم با عناصر طول ثابت (ادامه)
- بخش نشانه گذاری زباله روب کار دشواری است .
- سه فرضیه در مورد این فرآیند نشانه گذاری وجود دارد:
 - هر عنصر فعال باید از طریق زنجیره ای از اشاره گرها با شروع از خارج هرم قابل دستیابی باشد.
 - باید بتوان اشاره گرهای خارج از هرم را که به عنصری در هرم اشاره می کند تعیین کرد.
 - باید بتوان در داخل هر عنصر فعال هرم فیلدهایی را تعیین کرد که حاوی اشاره گرهایی به عناصر دیگر هرم اند.

مدیریت حافظه هرم (ادامه)

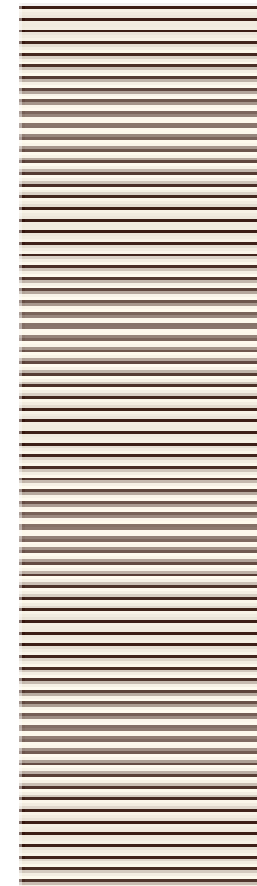


- مدیریت حافظه هرم با عناصر طول متغیر
- تخصیص اولیه و استفاده مجدد
- به دلیل متغیر بودن طول عناصر دو امکان برای استفاده مجدد وجود دارد:
 - استفاده از لیست فضای آزاد برای تخصیص حافظه
 - با انتقال تمام عناصر فعال به یک طرف هرم فضای آزاد لیست را فشرده کنید.

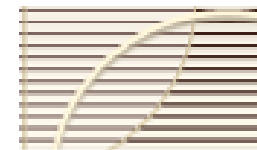
مدیریت حافظه هرم (ادامه)



- مدیریت حافظه هرم با عناصر طول متغیر
- استفاده مجدد از لیست فضای آزاد
- برای مدیریت تخصیص مستقیم از این نوع لیست فضای آزاد چند تکنیک وجود دارد:
- روش اولین جای مناسب
- روش بهترین جای مناسب

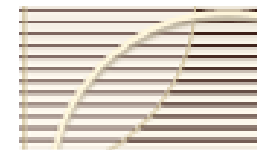


مدیریت حافظه هرم (ادامه)



- مدیریت حافظه هرم با عناصر طول متغیر
- بازیابی با بلوکهای طول متغیر
- روش برگشت صریح فضای آزاد شده به لیست فضای آزاد ساده ترین تکنیک است اما مسئله های مربوط به زباله ها و ارجاعهای معلق وجود دارد.
- باید تشخیص دهیم که انتهای یک عنصر کجاست و عنصر بعدی از کجا شروع می شود.
- ساده ترین راه حل این است که در کنار بیت زباله روبی در اولین کلمه هر بلوک طول آن بلوک نگهداری شود.

مدیریت حافظه هرم (ادامه)



- مدیریت حافظه هرم با عناصر طول متغیر
- فشردگی سازی و پراکندگی حافظه
- دو روش برای فشردگی سازی وجود دارد:
- فشردگی سازی جزئی
- فشردگی سازی کامل